



Model representation and documentation in computer simulation.

PIYASENA, Samarakoon M.

Available from the Sheffield Hallam University Research Archive (SHURA) at:

<http://shura.shu.ac.uk/20234/>

A Sheffield Hallam University thesis

This thesis is protected by copyright which belongs to the author.

The content must not be changed in any way or sold commercially in any format or medium without the formal permission of the author.

When referring to this work, full bibliographic details including the author, title, awarding institution and date of the thesis must be given.

Please visit <http://shura.shu.ac.uk/20234/> and <http://shura.shu.ac.uk/information.html> for further details about copyright and re-use permissions.

SHEFFIELD HALLAM UNIVERSITY
LEARNING CENTRE
CITY CAMPUS, FOND STREET,
SHEFFIELD S1 1WB.



Fines are charged at 50p per hour

REFERENCE

ProQuest Number: 10700879

All rights reserved

INFORMATION TO ALL USERS

The quality of this reproduction is dependent upon the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if material had to be removed, a note will indicate the deletion.



ProQuest 10700879

Published by ProQuest LLC (2017). Copyright of the Dissertation is held by the Author.

All rights reserved.

This work is protected against unauthorized copying under Title 17, United States Code
Microform Edition © ProQuest LLC.

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 – 1346

Model Representation and Documentation in Computer Simulation

Samarakoon M. Piyasena

**A thesis submitted in partial fulfilment of the requirements of
Sheffield Hallam University
for the degree of Doctor of Philosophy**



April 2004

ABSTRACT

Typically, simulation project is highly complex process, which relies heavily on the expertise and knowledge of the simulation analyst. It also requires the research of large amounts of systems data. This comprehensive data, together with the specialist skills of the analyst is integral to the success of any simulation project and it would seem obvious that a record of this information is ideally required for future references. However, it appears that, usually, very little or no effort is taken to record and maintain this significant information. This oversight often removes the opportunity for the subsequent use of the model by members of the project team themselves. It also hinders the reuse of simulation models in the development of future models that could use the same data. Hence, proper and complete documentation is seen as an essential requirement to overcome such situations.

A simulation study involves, not only developing the model, but also managing the process prior to model construction and subsequent tasks. Documentation in simulation involves, not only recording the model description, but also other exhaustive details embraced with the whole project. Clearly, the project team and model re-users are benefited from such in-depth and effective documentation.

Model Representation and Documentation (MRD) is a new concept for documentation in simulation. It addresses the different purposes or needs of different audiences in respect of the simulation project, model reuse, and other interested parties. No structured documentation methodology, either to satisfy this context, or to encompass the complete simulation project has been found in existing literature, or in simulation software. However, it is feasible that a progressive documentation with the model development process would fulfil the needs of different audiences and allows structuring the documentation process.

The proposed MRD process is based on task-orientation, which is attributed to the system development methodology in software engineering. It offers the user the ability to manage the documentation process with micro-level of task documents and to capture project details as the project progresses. Subsequently, task documents are accumulated to produce complete documents to fulfil different purposes of documentation. Pre-structured forms of task documents, which are based on typical simulation project procedure and enriched with reusable model elements, not only provide the uniform and consistent structure to capture task details, but also offer a sound foundation for an integrated documentation system.

An isolated MRD process, though concurrent with the model development, does not improve the present poor attempt to documentation. The integration of MRD process with model development offers the user the ability to perform both processes simultaneously as a single process while both are benefited directly and mutually through model exchange. The documentation models, which are constructed with reusable generic model elements, and the common database, which stores model details within a standard internal structure, make provision for such model exchange. Hence, an integrated MRD process improves, not only the documentation in simulation, but also model reusability.

The study has produced a novel approach for documenting the details of simulation projects in an integrated environment.

Samarakoon M. Piyasena

PREFACE

This thesis is submitted to the School of Engineering of Sheffield Hallam University for the degree of Doctor of Philosophy.

Except where acknowledge and reference are appropriately made, this work and the results obtained during the course of this research are, to the best of my knowledge, original and have been carried out independently. No part of the thesis has been, or is currently being, submitted for any degree or diploma at this university or any other university.

Samarakoon M. Piyasena

April 2004

ACKNOWLEDGEMENT

I would like to express my deepest gratitude and appreciation to my supervisor Prof. Terrence Perera (Sheffield Hallam University) for his guidance and constructive criticism during the course of study. I would like to extend the same gratitude to my second supervisor Dr. K.N.H.P. Liyanage (The Open University of Sri Lanka) for his help and assistance given by distance-learning mode.

I would also like to thank my colleagues and teaching and administrative staff of the School of Engineering and the research office for their help and support. The appreciation is extended to the Asian Development Bank and the Open University for awarding the scholarship, with which I was able to undertake this research programme.

Last, but not least, I would like to express sincere thanks to my wife Sunitha Samarakoon for her unfailing support and continuous encouragement. My children deserve much appreciation from me for their immense support with day-today domestic activities and for learning independence at such early stage.

TABLE OF CONTENTS

Abstract	i
Preface	ii
Acknowledgement	iii
Table of Contents	iv
1 Introduction	1
1.1 Background	1
1.2 Need for the Research	2
1.3 Focus of the Research	3
1.4 Outline of the Thesis	4
1.4.1 Chapter One: Introduction	4
1.4.2 Chapter Two: Literature Review	4
1.4.3 Chapter Three: Methodology and Establishing Framework	5
1.4.4 Chapter Four: Task Documents	5
1.4.5 Chapter Five: Presentation of Task Documents	5
1.4.6 Chapter Six: MRD Framework and Tool	5
1.4.7 Chapter Seven: Validation of the MRD Methodology	5
1.4.8 Chapter Eight: Discussion, Contribution Conclusion & Further Work	6
1.5 Scope and Key Assumptions	6
1.6 Summary	6
2 Literature Review	7
2.1 Introduction	7
2.2 Simulation Project/Project Life Cycle	7
2.2.1 Different Methods for a Simulation Project	7
2.2.1.1 Robinson's Procedure	8
2.2.1.2 Bank's Procedure	9
2.2.1.3 Pidd's Procedure	10
2.2.1.4 Law and Kelton's Procedure	11
2.2.1.5 Tye's Procedure	12
2.2.1.6 Shannon's Procedure	13
2.2.1.7 Maria's Procedure	14
2.2.1.8 Balci's Procedure	15
2.2.2 Analysis of Different Procedures in Simulation Project	16
2.3 Computer Simulation Tools	17

2.3.1	Classification of Simulation Tools	17
2.3.2	Analysis of Simulation Tools	18
2.3.2.1	Traditional General-Purpose Programming Languages	18
2.3.2.2	Simulation Software Packages	19
2.3.3	Different Styles in Simulation Software Packages	20
2.3.3.1	Traditional Paradigm in Simulation	20
2.3.3.2	Object-Oriented (OO) Paradigm in Simulation	20
2.3.3.3	OO Environment on Traditional Paradigm	21
2.3.4	Integration Interfaces in Simulation Software Packages	21
2.4	Reuse in Simulation	22
2.4.1	Reuse Spectrum	22
2.4.2	Components/Elements Reuse in Simulation	23
2.4.2.1	Reuse Components in Simulation Software	23
2.4.2.2	Building Reusable Components	23
2.4.2.3	Separation of Model Elements for Model Reusability	25
2.5	Model Representation (MR)	29
2.5.1	Classification of MR Techniques	30
2.5.2	Comparison of MR Techniques	30
2.5.2.1	IDEF Tools	31
2.5.2.2	Activity Cycle Diagrams (ACDs)	32
2.5.2.3	Petri Net (PN)	33
2.5.2.4	Unified Modelling Language (UML)	33
2.5.2.5	Flowcharts	34
2.5.3	Flowcharts for Model Representation and Documentation	36
2.6	Documentation in Simulation	37
2.6.1	Different Perspectives for Documentation in Simulation	38
2.6.2	Benefits from Documentation in Simulation	39
2.6.3	Current Practice in Documentation in Simulation	40
2.6.4	Documentation Approaches in Simulation	41
2.6.4.1	Conical Methodology for Documentation	41
2.6.4.2	Model Documentation with UML	43
2.6.5	Documentation Capabilities in Existing Simulation Software	43
2.7	Documentation in Software Development Process (SDP)	44
2.7.1	Software Development Process (SDP)/Software Life Cycle	44
2.7.2	Milestones in Documentation in SDP	45
2.8	Documentation in Other Modelling Areas	46

2.9	General Documentation Attributes	47
2.10	Summary	49
3	Research Methodology and Establishing Documentation Framework	50
3.1	Introduction	50
3.2	Research Questions	50
3.3	Research Methodology	51
3.3.1	Research Approach	52
3.3.2	Research Strategy	53
3.3.2.1	Grounded Theory for Data Collection	54
3.3.2.2	Action Research	54
3.3.2.3	Questionnaire Survey for Validation	54
3.3.2.4	Case Study for Supplementing the Questionnaire	55
3.3.3	Steps in the Research Process	55
3.4	Establishing Documentation Framework	58
3.4.1	Characteristics of Documentation Framework	58
3.4.1.1	Benefiting Audiences	58
3.4.1.2	Progressive Documentation	61
3.4.1.3	Continuous Recording	61
3.4.1.4	Model Representation as a Core of Documentation	62
3.4.1.5	Software Independent Documentation	62
3.4.1.6	Integrated Documentation	62
3.4.1.7	Structured Approach	63
3.4.2	Research Question 01: What is Documentation in Simulation?	63
3.4.2.1	Definition for Model Representation and Documentation	63
3.4.2.2	Task-Orientation	64
3.4.3	Revised Project Life Cycle for MRD in Simulation	66
3.5	Summary	68
4	Task Documents	69
4.1	Introduction	69
4.2	Primary Tasks in a Simulation Project	69
4.3	Task Documents/Deliverables in Primary Tasks	71
4.4	Task Documents in Phase 01: Project Specifications	71
4.4.1	Problem Formulation	72
4.4.1.1	Project Communication	72
4.4.1.2	Problem Analysis	73

4.4.2	Problem Definition	73
4.4.2.1	Objectives Identification	74
4.4.2.2	Experiments and Reports Identification	75
4.4.3	System Investigation	76
4.4.3.1	Process Analysis	76
4.4.3.2	Model Scope and Level Identification	77
4.4.4	Model Formulation	80
4.4.4.1	Model Scope and Level Reduction	80
4.4.4.2	Model Splitting	81
4.4.5	Data Acquisition	82
4.4.5.1	Data Collection and Estimation	82
4.4.6	Project Validation	90
4.4.6.1	Project Summarization	90
4.4.6.2	Model Summarization	91
4.5	Task Documents in Phase 02: Model Building and Testing	92
4.5.1	Model Representation	93
4.5.1.1	Model Representation for Documentation	93
4.5.1.2	Model Structuring	96
4.5.1.3	Data Capturing	96
4.5.2	Model Coding and Verification	97
4.5.2.1	Model Coding and Documentation	97
4.5.2.2	Model Verification	98
4.5.3	Model Validation	99
4.5.3.1	Operation Validation	99
4.5.3.2	Parameter Validation: Sensitivity Analysis	100
4.6	Task Documents in Phase 03: Experimentation	100
4.6.1	Design of Experiments	101
4.6.1.1	Design of Strategic Experimental Plan	101
4.6.1.2	Design of Tactical Experimental Plan	102
4.6.2	Experiment Run and Results Analysis	103
4.6.2.1	Experiment Run	103
4.6.2.2	Results Analysis	104
4.7	Task Documents in Phase 04: Project Implementation	105
4.7.1	Result Communication	105
4.7.1.1	Reviewing the Results	106
4.7.1.2	Reviewing the Project	106

4.7.2	Project Completion	107
4.7.2.1	Development of Implementation Plan	107
4.7.2.2	Completion of Documents	108
4.8	Summary	112
5	Presentation of Task Documents	113
5.1	Introduction	113
5.2	Analysis of Task Documents in Phase 01: Project Specifications	114
5.2.1	Overview of the Client	114
5.2.2	Problem and Need Statement	114
5.2.3	Project Team	115
5.2.4	Form of Review	116
5.2.5	Review Outcomes	117
5.2.6	Problem Objectives	118
5.2.7	Model Review Objectives	119
5.2.8	Work Plan Objectives	120
5.2.9	Future Use of the Model	121
5.2.10	Experiment Details	122
5.2.11	Report Details	123
5.2.12	System Description	124
5.2.13	Physical Elements and Details to be Modelled	125
5.2.14	Simplified Elements and Details	126
5.2.15	Excluded Elements and Details	127
5.2.16	Subsystems	128
5.2.17	Model Layout	129
5.2.18	Data Description	130
5.2.19	Data Omitted	132
5.2.20	Validity of Project Specifications	132
5.2.21	Dormant Changes in Project Specifications	133
5.2.22	Validity of Model Specifications	134
5.2.23	Additional Time-scale and Milestone	135
5.3	Analysis of Task Documents in Phase 02: Model Building and Testing	135
5.3.1	Documentation Model	135
5.3.2	Model Data	139
5.3.3	Model Details	141
5.3.4	Verification Details	142
5.3.5	Details in Operational Validity	143

5.3.6	Details in Parameter Validity (Sensitivity Analysis)	143
5.4	Analysis of Task Documents in Phase 03: Experimentation	144
5.4.1	Overview on Experimentation	144
5.4.2	Warm-up Period (WUP) and Starting Conditions	145
5.4.3	Run Length	146
5.4.4	Number of Replications	147
5.4.5	Summary of Experiment Run	148
5.4.6	Summary of Results	149
5.4.7	Conclusions and Recommendations	150
5.5	Analysis of Task Documents in Phase 04: Project Implementation	151
5.5.1	Review of the Results	151
5.5.2	Review of the Project	152
5.5.3	Implementation Plan	153
5.5.4	Documents Prepared	154
5.6	Summary	155
6	Model Representation and Documentation (MRD) Framework and Tool	156
6.1	Introduction	156
6.2	MRD Process	156
6.3	MRD System	157
6.4	Automated MRD System	158
6.4.1	MRD Subsystems	159
6.4.1.1	Project Specification Subsystem	160
6.4.1.2	Model Building and Testing Subsystem	161
6.4.1.3	Experimentation Subsystem	162
6.4.1.4	Implementation Subsystem	162
6.4.2	Documentation Model (DM) in the Core of MRD	163
6.4.2.1	VISIO to Create DM and DM Database	164
6.4.2.2	DM Template	164
6.4.2.3	DM Stencil	166
6.5	MRD Tool for On-Line Documentation	167
6.5.1	Interface for Handling Task Documents	168
6.5.1.1	Tree View of Task Documents	169
6.5.1.2	Check Boxes in Tree View	170
6.5.1.3	Highlights of Task Documents in Tree View	170
6.5.1.4	Immediate Display Screen for Task Documents	170
6.5.2	WORD-VISIO Interface	171

6.5.3	VISIO-ACCESS Interface	172
6.5.4	WORD-ACCESS Interaction	174
6.5.5	User Interface for Documents Preparation	176
6.6	Model Exchange	175
6.6.1	Model Exchange with ARENA	177
6.6.1.1	Functions and Limitations in Pre and Post Documentation	178
6.6.1.2	Mapping Tables	181
6.6.2	Model Exchange with Other Simulation Software	184
6.8	Summary	184
7	Validation of the Proposed Methodology	185
7.1	Introduction	185
7.2	Questionnaire	185
7.2.1	Specific Objectives of the Questionnaire	186
7.2.2	Format of the Questionnaire	187
7.3	Case Study	188
7.4	Participants	188
7.5	Survey Results and Analysis	188
7.5.1	Current Practice and General Opinion	189
7.5.2	How Effective the MRD Process?	190
7.5.3	How does the MRD System Assist?	191
7.5.4	How Effective the Integrated MRD System?	192
7.5.5	Does the Methodology Offer Other Benefits?	193
7.5.6	Weaknesses and Limitations of the Methodology	193
7.6	Validity of the Proposed Methodology	195
7.7	Summary	195
8	Discussion, Contribution, Conclusion and Further Work	196
8.1	Introduction	196
8.2	Discussion	196
8.2.1	Research Question 01 and Model Representation & Documentation	196
8.2.2	Research Question 02 and Task Documents	197
8.2.3	Research Question 03 and Integrated MRD System	199
8.2.3.1	Uniform Structure	200
8.2.3.2	Sharing Data	201
8.2.3.3	Exchanging the Model	202
8.2.3.4	Interfacing with the User	203

8.2.3.5	Limitations and Weaknesses in MRD System	204
8.2.3.6	MRD System as a Stand-Alone or Remote System	204
8.2.4	Additional Benefits from MRD System	204
8.2.5	MRD System or Standard MRD System	205
8.3	Contribution	207
8.4	Conclusion	208
8.5	Further Work	209
References		210
Annexure A - Mapping Tables		A1
Annexure B – Questionnaire Survey: Validation of the Proposed Methodology		B1
Annexure C - Case Study: An Example for Simulation Project Document		C1
Annexure D – Summary of the Questionnaire Survey Results		D1

1. INTRODUCTION

1.1 Background

Simulation has become a widely used indispensable tool in the decision-making process in a diversity of fields in real-world applications. Since its inception in the 1950s, it has been used to solve the wide range of problems leading to efficiency enhancement, cost reduction and profit advancement in business systems, manufacturing industries, service sectors, communication networks, military applications and many more. It is well established in manufacturing industry as an integral tool for the designing, planning, operations and the maintenance of manufacturing systems, though, traditionally, it was mainly used for capital-intensive projects such as designing new factory layouts.

Simulation is the imitation of the operation of a system or process over time. A simulation model, as it evolves, studies the behaviour of the system. Discrete event simulation is concerned with studying a model of the system, in which, the behaviour changes with time at sudden distinct events. Manufacturing reflects the characteristics of both discrete and continuous simulation, but most manufacturing systems can easily and adequately be studied with discrete event simulation models. As the major focus of this research is in respect of manufacturing applications, discrete event simulation is the major concern. Hence, 'simulation' refers to 'discrete event simulation' in this thesis.

The discrete event simulation process typically consists of a number of different procedural steps, which are executed sequentially or iteratively. The entire process, i.e. the simulation project, generally has complex and lengthy procedural steps. It begins with the analysing of observations in a real-world system and then formulating a logical model of the system for the development of a computer model. The computer model enables the user to experiment for the purpose of studying the system and obtaining the decision supporting information. At each of these transformation steps, the modellers bring the initial system into a simplified decision-supporting model by analysing the background information of the system through their knowledge and experience. In the process of system simplification or bringing the problem into sound solution, the project team encounters vital information such as system data and decisions influenced by several assumptions about the model and the background information. This information plays an important role, not only in the current project, but also in the future use of the model for successful understanding, building, altering, experimenting, analysing and implementing the model. Therefore, recording the details of simulation projects for the benefit of current project and the future usage of the model has become a challenging topic in the simulation field. Therefore, this study focuses on the documentation of details that are involved in simulation projects.

1.2 Need for the Research

The increasing complexity of systems has enhanced the use of simulation models for the decision-making process. Hence, simulation is gaining increasing importance in real-world applications. However, the effective use of simulation models has been hindered due to many reasons such as poor conceptualisation, inefficient data handling, insufficient model verification and validation, poorly planned experimentation, poor documentation and unstructured implementation (Liyanage, 1999). Much research has focused on most of the above issues. However, no great attempt has so far been made for recording the details of simulation projects in order to keep the track on how the model was built, enabling the model to be understood, updated, re-used, and inherited by others, although benefits of documentation spreading throughout the project are greatly appreciated.

The simulation project is generally more than building a model; it is a managing process, working with several people and handling a large amount of information and broad technical knowledge to achieve a certain goal. In such an environment, proper communication through a well-documented procedure avoids early misunderstanding, minimizes project delays, and assists to ensure the validity and success of the project. Therefore, it appears that the sooner the documentation is dealt with, the faster the simulation project development would be. Hence, the documentation should be an integral part of the simulation process.

Simulation usually requires taking a large number of different alternatives and parameters into account in system realization and model design. Variations in design and the information involved in one model to another are often relatively small in systems like flexible manufacturing systems where similar structure and layouts are kept on focus. However, it appears that the model and knowledge acquired in one simulation project are only used in another project to a minor extent. Often, all the efforts and costs incurred during the simulation models, which may have subsequent usage, are forgotten once the results of the simulation are determined and therefore new models are built from scratch. It seems that one of the major reasons for such a situation is due to none or poor recording of details of the model and its background. In such circumstances, model modification or model reuse is extremely difficult without the benefits of the details that went into constructing the model. If the model is expected to be reused without recorded project details, reverse engineering of the computer model will be the only way of understanding the underlying logic of the model. Therefore, subsequent modellers, who wish to reuse the model, may have to 'reinvent the wheel' again and again by spending time and resources repeatedly and unnecessarily.

Despite the great deal of cost and time spent in finding the most elegant model to represent the system, it seems that modellers throw the model to the client at the last minute, without recording useful details of the project. However, whenever the documentation is attempted, the modellers record details idiosyncratically and in isolation. Such situations will result an ill-defined final model and results with varying degree of interpretation. Usage of such simulation models cannot be effective, unless all the details of the model are clearly and concisely recorded, enabling the user to understand the model and results precisely. This clearly concludes the significance of documentation in simulation, as well as a need for a structured methodology for documentation of simulation projects to ensure the success of simulation projects and for enhancement of models reuse. Such a requirement has been emphasized by the comment made by Rio (1999) saying that lack of documentation methodology in simulation had also slowed down the spreading of simulation software.

1.3 Focus of the Research

The existence of an appropriate documentation for an entire simulation project life cycle is one of the important factors for the success of simulation projects, as well as model reuse. However, the modellers have given varying attempt for this time-consuming documentation task due to their lack of a systematic approach for recording simulation project details. The preliminary question that arises now is **‘how should the documentation of simulation projects be accomplished to ensure the success of simulation projects and model reuse?’** Focus on this question led to the formulation of the research title -

“Model Representation and Documentation in Computer Simulation”,

- with the aim of **developing a methodology to document simulation projects in manufacturing applications.** In order to find answers to the above target question and accomplish the aim of the research, the followings objectives were set.

1. Reviewing the concept and needs in documentation in order to refine the context of documentation in simulation.
2. Reviewing the existing documentation:
 - methodologies in simulation model development process,
 - capabilities in current simulation software,
 - approaches in other modelling process (e.g. Ecological modelling),
 - practice in software development process, and
 - attributes in general documentation.

in order to identify their features, strengths and limitations and to establish the documentation framework.

3. Analysing the existing simulation project procedures and model development process in a reusable form in order to:
 - identify the details that need to be documented, and
 - recognize the needs of audiences that are benefited from documentation.
4. Constructing a new framework that is based on the above reviews and analysis stages so that documentation is performed independently from simulation software and concurrently with simulation model development.
5. Developing interfaces for on-line documentation for new or existing projects.
6. Synthesizing the framework and tool with a commercial simulation software for an integrated documentation system.
7. Validating the proposed documentation methodology.

On meeting the above objectives, it is expected that the documentation of simulation projects would be popular and an un-isolated function in simulation.

1.4 Outline of the Thesis

The thesis comprised of eight chapters. Each chapter, which starts with a brief introduction to the contents and ends with concluding remarks on proceeding, describes the fundamental component of the research programme, data, data analysis process and results of the analysis. The chapters are written as separate pieces of work, wherever possible, independently from other chapters. In order to present them as independent, forward or backward references are made wherever necessary so that the reader can trace the necessary facts and information easily and quickly.

1.4.1 Chapter One: Introduction

This Chapter begins with explaining the background and justification of the research. It furnishes how the research was focussed chronologically for meeting the aim of the study. The overview of each chapter was then presented for referencing the contents of the thesis. It concluded by declaring the scope and key assumptions made in the study and the thesis.

1.4.2 Chapter Two: Literature Review

Chapter two reviews the literature of the subject area in simulation in general; and discusses and compares the strength and weaknesses of different approaches in simulation project life cycle, model reuse, model representation and documentation in simulation, as well as other allied areas. While the subject materials are being reviewed, the study is narrowed down by setting focus on the reference material. Key issues that were raised during the review are highlighted to extract the research questions that drive the research process for the development of a rational framework.

1.4.3 Chapter Three: Methodology and Establishing Framework

This chapter presents the major research questions that need to be addressed and which research approach was selected to address those questions and why. The choice of the strategy or strategies to achieve the objectives of the research question is justified with strengths and weaknesses of research approaches and strategies in general, followed by a discussion on the research steps, which immersed in answering the questions. It also presents how the initial documentation framework was established, based on the findings of the literature survey and the general views that were obtained from simulation practitioners.

1.4.4 Chapter Four: Task Documents

This chapter addresses the second research question, the objective of which is to identify the details that need to be documented in a complete simulation project. For this purpose, the investigation process performed on the revised simulation project procedure is briefly presented and the results of the investigation are listed as task documents. The chapter concludes by recognizing task documents that fulfil different needs of different audiences.

1.4.5 Chapter Five: Presentation of Task Documents

Having produced the list of task documents, this chapter presents an overview of each of them, highlighting the contents in generic and reusable form, followed by a discourse of the significance of each document in respect of different audiences and the purposes of documentation. Structure and layout to record details of each task document is then presented by maintaining the consistency and uniformity of records, wherever possible.

1.4.6 Chapter Six: MRD Framework and Tool

This chapter describes how the framework is developed for recording the details, which have been identified previously, in concurrent with the model development process. The user-interfaces developed for handling subsystems in different software, storing and retrieving information, final document preparation and model exchange for on-line documentation system is discussed in detail. The task related to model representation will also be elaborated in great extent, as it is a vital component in integrated MRD system.

1.4.7 Chapter Seven: Validation of the Proposed Methodology

The validation process, which is supported by a questionnaire and an illustration of a hypothetical case study to demonstrate the application of the proposed methodology for the production of document of a simulation project, is discussed in this chapter. The findings of the validation process is dealt with in great detail in order to substantiate the validity of the proposed methodology in respect of viability and practicability in real-world applications and the feasibility of synthesizing the framework with commercial simulation software.

1.4.8 Chapter Eight: Discussion, Contribution, Conclusion and Further Work

This chapter discusses the proposed methodology for MRD in the light of objectives set against the research questions. In appropriate points in the discussion, advantages, and disadvantages of the proposed approaches or techniques, limitations of the findings, and additional benefits are conveyed. Subsequently, avenues for further research are proposed. The chapter sums up by presenting the contribution made to the knowledge of simulation, followed by the conclusion of the research.

1.5 Scope and Key Assumptions

- The research is restricted to discrete-event simulation as applicable to manufacturing applications.
- An appropriate simulation project life cycle was not selected for identification of details that need to be documented (task documentation). Instead, a revised project procedure that is based on existing project procedures was followed for task documentation.
- Suitability of the flow charts technique for model representation task was not validated. Instead, its appropriateness for documentation was reviewed with existing literature.
- The references are not indicated in analysis process of task documents and task details (in Chapters 04 and 05) with the assumption that they are very common. They are mainly based on Robinson (1994), Banks (2000), Pidd (1989, and 1992), Law and Kelton (1991), Tye (1999), Shannon (1998), Maria (1997), and Balci (1998), unless otherwise stated in appropriate places.
- It is assumed that the reader is familiar with basic technical aspects on simulation.

1.6 Summary

The significance of recording the details of simulation projects was emphasized and the appropriateness of the study in documentation in simulation was justified in respect of the simulation project and model reuse through different views presented in literature and gathered information from the simulation community. Then, the objectives were set to accomplish the aim of the research topic - model representation and documentation in computer simulation. The brief overview of each chapter and the scope, together with key assumptions made in the research and the thesis, were presented to guide the reader.

2. LITERATURE REVIEW

2.1 Introduction

The literature review in this study is wide-ranging as the scope of the study spreads throughout the entire simulation project. An extensive literature review was conducted in the past and the present developments in simulation and other related areas, such as software development. At the end of each major review stage, any choice made and key issues raised are emphasized for keeping the focus on the study.

With the assumption that the reader is familiar with the history and the terminology in basic simulation concepts, this chapter begins with presenting common documented procedures of simulation projects. A review of procedures, with the focus of identifying different stages for the purpose of documentation, is then followed. Model representation is a key element in the documentation of simulation. Hence, existing model representation techniques, different approaches in generation and representation of reusable elements, and the effect of different simulation software architecture on reusable elements are then reviewed for making a choice for appropriate reusable and software-independent components. Finally, existing approaches and capabilities in the documentation of simulation projects, in simulation software, in software development process and in other areas related to simulation are reviewed. The chapter concludes by describing the focus of the study from the gathered information and the key issues identified during the literature review.

2.2 Simulation Project/Project Life Cycle

A simulation project or project life cycle is generally a complex, lengthy, and iterative process involving different stages. Each stage involves different processes, produces different results, and involves many details. Identification of this information is vital for the development of a documentation framework. Therefore, the review begins with analysing the process of the generic simulation project life cycle and then identifying the details that are to be recorded. However, identifying the details has faced a major hindrance due to the inconsistency of the breakdown of the project procedure proposed by different authors.

2.2.1 Different Methods for a Simulation Project

The method of conducting a simulation project is documented by different authors from different perspectives and using varying terminologies, but the conceptual approach remains the same in all procedures. Therefore, this section discusses and compares the similarities and differences between various procedures, followed by a review of major activities in different stages in order to agree with an existing procedure or to propose a procedure to ensure documentation, which would be acceptable as generic. This discussion is centred around only the major activities and details involved with those activities.

2.2.1.1 Robinson's Procedure

Robinson (1994) presented a general overview of a simulation project, consisting of four major phases as in Fig. 2.1. The author highlighted that the study should always start from problem definition and move towards the project implementation. Despite the fact that it has been shown in linear fashion, the author stresses that the entire process is an iterative process and the iteration is illustrated by the upward arrows. In addition, each phase is further broken-down into smaller steps and they are represented in both sequential and iterative nature. Table 2.1 below summarizes the tasks to be performed in each of those phases, as they are helpful in deciding the project procedure for the purpose of documentation and the identification of details to be documented.

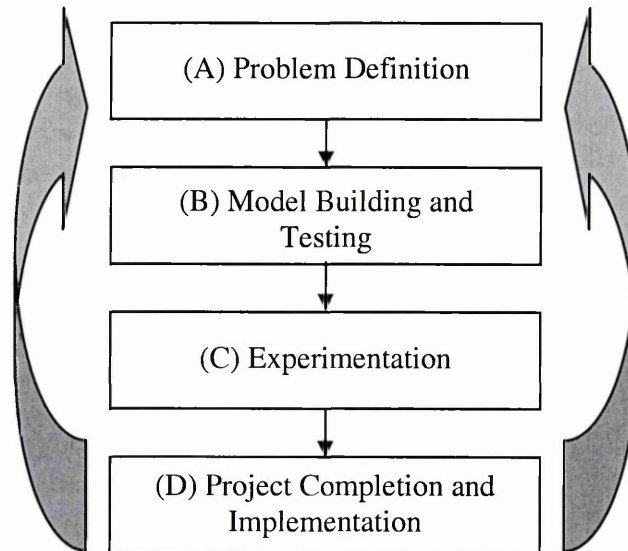


Fig. 2.1 : Steps in Simulation Projects [Robinson, 1994]

Steps	Major Tasks
A	Identify the problem and possible solutions; set problem and general objectives; suggest additional objectives; rank objectives; discover future uses of the model; decide the experimental factors, likely range of values & the data entry methods; identify the values to be reported & the method of reporting and viewing; define the scope and level of the model; reduce the scope and level (simplify the model); identify the data required for different purpose; decide how to deal with unobtainable and inaccurate data; select method of data entry; start the data gathering; structure & communicate project specification and decide how the changes are handled.
B	Structure overall model; divide the model into sections for coding; code & verify the model incrementally for sections and sub-models; decide the methods for model validation, validate the general objectives & the model; investigate and correct errors; test the sensitivity on unobtainable data.
C	Selects the experiments to be conducted, determine how the experiments are conducted; decide the warm-up period or starting conditions, identify run length or number of replications; perform the experiments; decide alternatives for results analysis for each set of data; compare the outcomes and draw conclusions.
D	Communicate the results, conclusions, & recommendations; identify who implement them; develop an implementation plan; implement the recommendations; complete the documentation; review the project; and perform further work.

Table 2.1 : Major Tasks in Simulation Projects [Robinson, 1994]

2.2.1.2 Bank's Procedure

Bank's (2000) procedure contains detailed description of the simulation project and concentrates the iterative nature of the process in details. He also addresses the time frame involved in different steps, for example, presenting model conceptualisation and data collection as a parallel process, and thus, in general, it provides more depth detail in the project life cycle. Twelve major steps involved with this approach are shown in Fig. 2.2 and their major tasks are briefly depicted in Table 2.2.

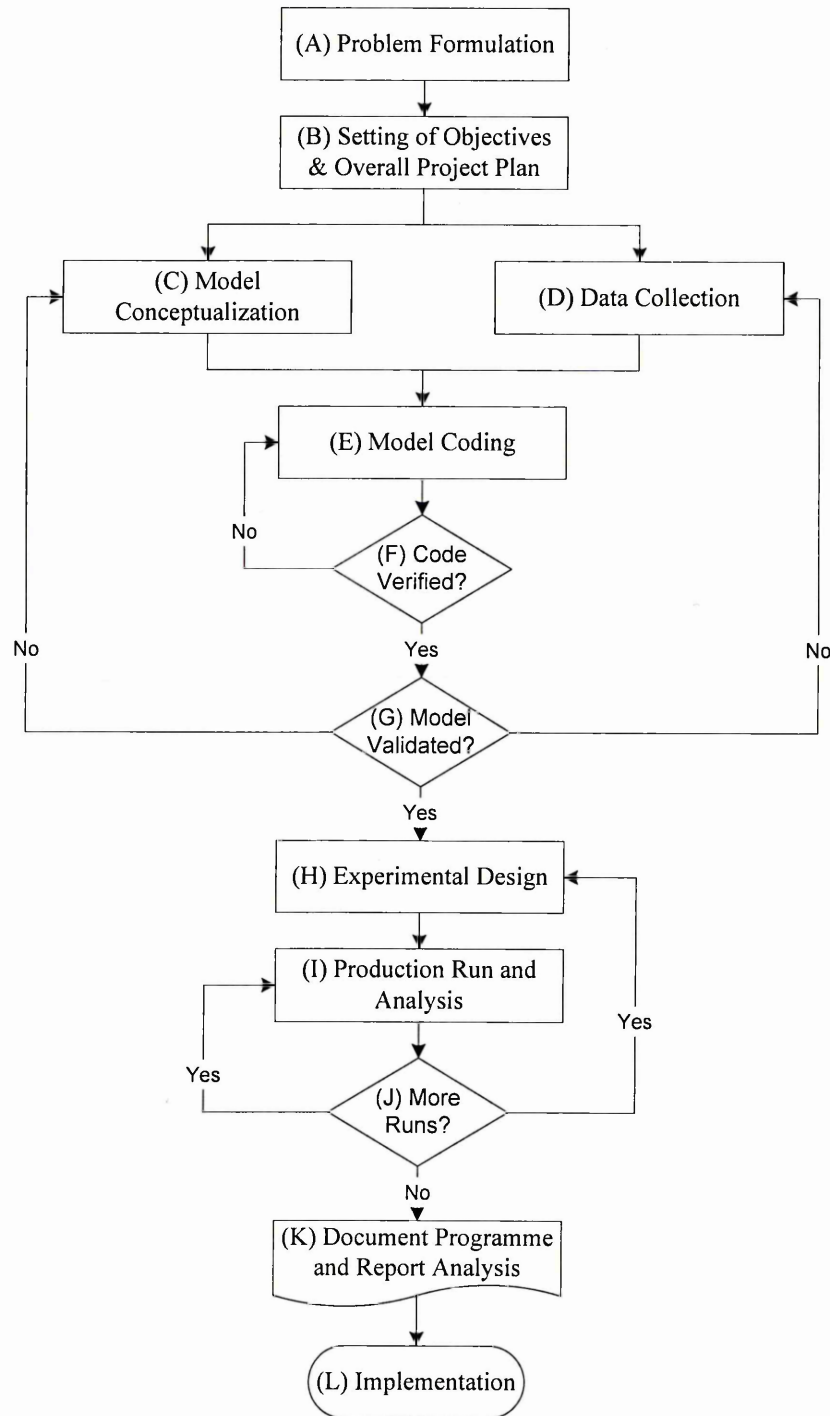


Fig. 2.2 : Steps in Simulation Projects [Banks, 2000]

Steps	Major Tasks
A	Provide the problem statement as symptoms; prepare the problem statement; agree with problem formulation and set & agree the possible assumptions.
B	Set the objectives; prepare the project plan in term of time, personnel, hardware, software, progress & investigation stages, output, cost & billing procedure.
C	Establish the mathematical and logical relationships of basic components; add complexity step by step; get the client involved to enhance the quality and confidence and get the approval on the conceptual model.
D	Identify the data and format required; assign the relevant personals for data collection and distinguish the available and non-available data.
E	Code the conceptual model into computer operational model while the data is being collected.
F	Choose the method of verification and verify the model while it is coding
G	Decide how the model is validated and validate the model for its objectives.
H	Decide the different scenarios and determine the length of simulation run, the replications, & initialisation conditions, if required.
I	Perform the experiments; analyse & estimate the measures of performance for each scenario and draw recommendations.
J	Review the results; determine the need of additional runs and simulate the model.
K	Record and communicate the results, recommendations, and alternatives.
L	Communicate the conclusions & recommendations; review recommendations; and implement the project.

Table 2.2 : Major Tasks in Simulation Projects [Banks, 2000]

2.2.1.3 Pidd's Procedure

In a number of publications, Pidd (1989, and 1992) presented three key phases of a simulation project, but the author assumes that the analyst and the client have already agreed which specific problems should be tackled, provided that the problem is feasible for a simulation project. He specifically argues that no two-simulation projects will be identical to be able to propose a common procedure, though the following three general steps are suggested.

A. Modelling

B. Computing

C. Experimentation

Pidd considers modelling and programming as preliminaries to the real purpose of simulation, that of experimentation. He strongly argues that, in practice, these three phases may be difficult to separate precisely. In this respect, with his experience, he constantly reiterates that experimentation often leads to changes in the model as well as in the computer programme. This concludes that three major phases are intimately linked. However, in this procedure, the above phases are not broken down into more details tasks. Instead, he highlights the importance of validation in the modelling phase and different approaches in the computing phase. In respect of the experimentation phase, the author emphasizes the need for a specific plan before experimenting with the model for different scenarios.

2.2.1.4 Law and Kelton's Procedure

One of the most cited approaches in literature is Law and Kelton's procedure. Law and Kelton (1991) proposed a series of ten steps, somewhat similar to the Bank's approach, indicating its logical sequence in a flowchart to represent the iterative nature of the simulation project. The major highlighted features in this procedure are that the data collection commences at an early stage of the project and that the validation is something that should continue through out the study. Fig. 2.3 shows the flowchart of the activities and Table 2.3 embodies the major tasks in each step.

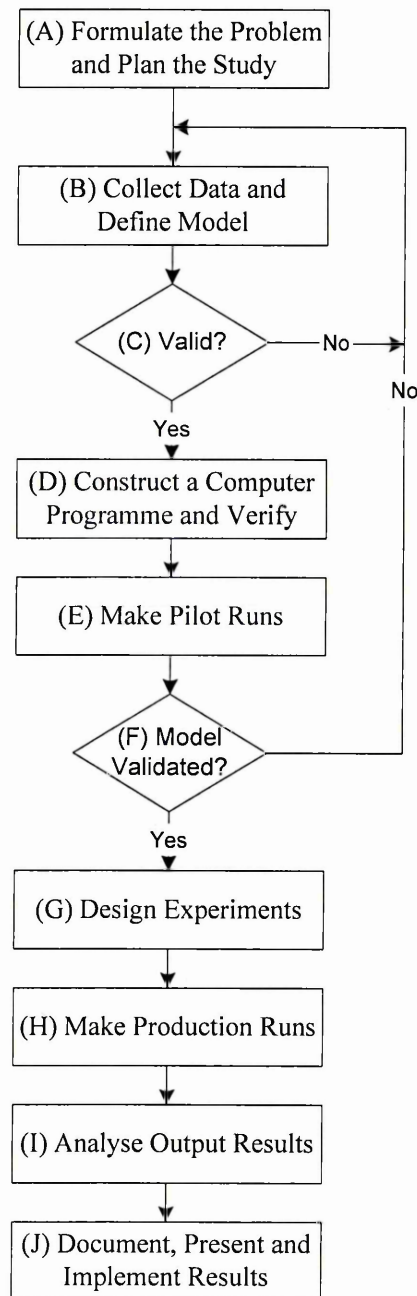


Fig. 2.3 : Steps in Simulation Projects [Law and Kelton, 1991]

Steps	Major Tasks
A	State clear statement on the overall objectives of the study & any specific issues to be addressed; define any alternative system design, if available; establish the criteria for evaluating alternatives; and plan the overall study in respect of people, cost & time.
B	Start the collection of data to specify operating procedures & probability distributions for random variables used in the model; extend the data collection for validation; define the model with moderate details to avoid excessive costs involved in programming, but allowing to capture sophisticated details in later stage, if required.
C	Validate the conceptual model with people who are involved with the actual system & decision makers on a regular basis; and test the probability distributions.
D	Decide the programming method, whether the general-purpose languages or simulation languages; programme or code the model and verify or debug the codes.
E	Make a pilot runs and obtain the output.
F	Conduct sensitivity analysis of input data; and compare output data values from a real system, if exists to validate the model.
G	Design experiments to evaluate the different system design, decide the types of experiments required; and define initial conditions, warm up period, length of simulation runs and number of replications appropriate for each alternative.
H	Run the model to generate performance data for each experiment.
I	Analyse the output data; construct the confidence interval to measure performance of different system design; and compare the data from different systems.
J	Document the assumptions & computer programme; and implement the model

Table 2.3 : Major Tasks in Simulation Projects [Law and Kelton, 1991]

2.2.1.5 Tye's Procedure

On a study conducted on the model design process, Tye (1999) concludes that a simulation project does not progress in a sequential manner and that it has a strong tendency towards a highly iterative nature. He also stresses that it is very difficult to separate stages of the project and that the modellers engage in cycles of activities rather than series of stages. He suggests that the progression of the project is characterised by four phases, similar to Robinson's procedure (1994), as shown in 2.4. The major tasks performed in these stages are listed in Table 2.4. The tasks in phases C and D were not concluded, but the relevant task was taken from the action research conducted by the author.

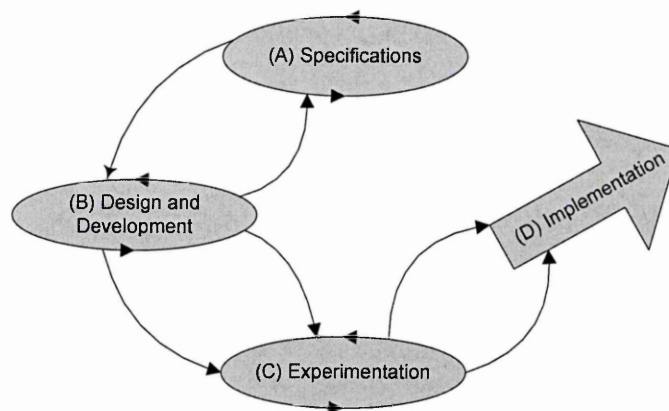


Fig. 2.4 : Steps in Simulation Projects [Tye, 1999]

Steps	Major Tasks
A	Choose the methods for problem formulation; identify the obstacles in formulation; understand the real problem; establish the objectives with performance indicators (i.e. measures) that give indicators on experiments, model scope & level.
B	Choose the methods for system investigation; determine the primary elements to be included in the model (avoid elements which make the model complex); select how system elements are represented in the model; determine how the model is simplified; break down the overall model into manageable “chunks” (sub-models); separate the logic from data; and code sub-models and the overall model.
C	Determine how the experiments are conducted; decide the experimental parameters; perform the experiments; compare outcomes and draw conclusions. (Not concluded).
D	Communicate the results, conclusions, & recommendations; implement the recommendations; complete documentation; and review the project. (Not concluded).

Table 2.4 : Major Tasks in Simulation Projects [Tve, 1999]

2.2.1.6 Shannon’s Procedure

Shannon (1998) proposed a procedure for simulation, merging the process with software development practice. In this approach, the author identified twelve major steps as listed in Table 2.5. He highlights the effectiveness of spending much time on the planning stage before starting the model translation stage. He compared this approach with the “40-20-40” rule used in software engineering, by which it is explained that 40% of time and effort should be devoted to steps A to F, 20% to step G and 40% for the remaining steps. The major drawback in this procedure seems to be that the data may not be available for the model-coding phase as the time-consuming data preparation task begins at a later stage.

Steps	Major Tasks
A. Problem Definition	Present the problem/situation as symptoms; diagnose the symptoms; define the goal; identify decisions to be made; recognize the information required to make decisions; determine how and who makes decisions.
B. Project Planning	Identify the support required from management; identify team members with their skills and responsibilities; recognize the available resources and constraints; and establish adequate communication stages.
C. System Definition	Determine the boundaries of the system; divide the system into logical subsystems; investigate how the processes in system & subsystem work.
D. Conceptual Model Formulation	Define entities, resources (stations), variables, flow patterns of entities, alternatives to the system, and develop flow charts for routing logic.
E. Preliminary Experimental Design	Define the measures of effectiveness; select the factors to be varied and identify what data to be gathered in what levels and what form.
F. Input Data Preparation	Identify the input data required; recognize the data sources, availability of pertinent data and validity of the data; and start gathering data.
G. Model Translation	Choose the appropriate software for translation and translate the conceptual model into a computer model.
H. Verification and Validation	Select how the computer model is verified; check the model for correctness, debug the model; select how the model is validated; assess accuracy and validity of the model; and alter the model if required.

Table 2.5 : Steps and Major Tasks in Simulation Projects [Shannon, 1998] (Contd..)

Steps	Major Tasks
I. Final Experimental Design	Change project plan to suit the remaining resources and constraints, alteration to experimental factors for the simplified model developed and decide run length to suit time constraint.
J. Experimentation	Decide starting conditions, warm-up period, run length, sampling size, number of replications, depending on the type of system, and conduct the experiments to generate the desired data.
K. Analysis and Interpretation	Analyse the data for inferences with the goal defined and interpret the results.
L. Implementation & Documentation	Report the results, implement the outcomes, record the findings, and document the models.

Table 2.5 : Steps and Major Tasks in Simulation Projects [Shannon, 1998]

2.2.1.7 Maria's Procedure

Maria (1997) suggested a general procedure somewhat similar to Shannon's, Law & Kelton's and Bank's approaches, comprising eleven phases. It is argued that much iteration at various sub-stages may be required before the objectives are achieved; and not all the steps may be possible or required, depending on the project. The major tasks performed in these stages are briefly listed in Table 2.6.

Steps	Major Tasks
A. Identify the Problem	Enumerate the problem and produce the best requirements for the proposed system with respect to the problem presented.
B. Formulate the Problem	Select the boundaries of the system; define the objectives with specific issues & performance measurements; rank objectives; make assumptions; decide the time frame of the model use (one-time or regular basis) and identify the end user.
C. Collect and Process Real System Data	Collect data on input variables and system specifications; identify the randomness of variables; select appropriate probability distribution for each variable and estimate parameters.
D. Formulate and Develop a Model	Develop schematic and network diagrams to represent entity flow; translate conceptual model into computer model; select appropriate methods for verification and alter the model as the coding progresses.
E. Validate the Model	Run the model to compare the model performance with known conditions; perform statistical tests and demonstrate the model.
F. Document Model for Future Use	Record objectives, assumptions, and input variables in detail.
G. Select Appropriate Experimental design	Select input variables that are likely to affect on each performance measure and document experimental design.
H. Establish Experimental Conditions for runs	Select how the information is required (accurate/most) and determine starting condition, warm up period, run length & number of runs, appropriately.
I. Perform Simulation Runs	Perform experimentation; collect data and observation and analyse them.
J. Interpret and Present Results	Compute numerical estimates; test hypothesis about the performance; draw conclusions and document them.
K. Recommend Further Course of Action	Decide on further experiments.

Table 2.6 : Steps and Major Tasks in Simulation Projects [Maria, 1997]

2.2.1.8 Balci's Procedure

Balci's (1998) procedure has gained widespread popularity in respect of model verification and validation. With the emphasis of validation and verification, the author describes the project with eight stages. It is also argued that the overall project life cycle is not a sequential process and that it can be reversed when errors occurred. Unlike in other procedures, the verification and validation process is extended throughout the entire simulation project procedure as represented by solid arrows in Fig. 2.5. Major tasks in this procedure are self-explanatory within the diagram.

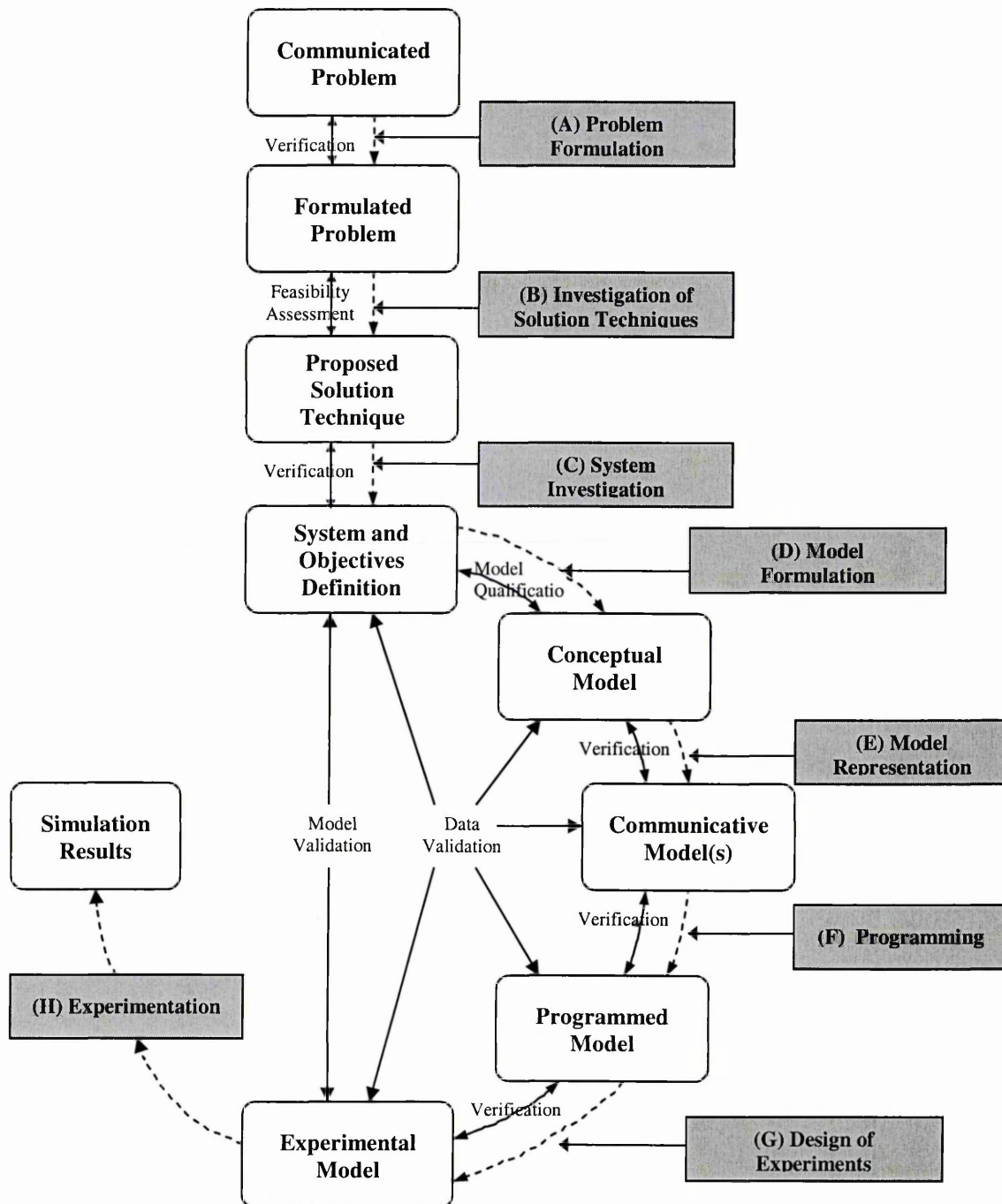


Fig. 2.5 : Steps in Simulation Projects [Balci, 1998]

2.2.2 Analysis of Different Procedures in Simulation Project

The analysis of different project procedures exemplifies that the breakdown of phases or stages in a simulation project comes in two perspectives, i.e. in broader aspects and detailed breakdown. Robinson (1994), Pidd (1989 and 1992) and Tye (1999) consider the project life cycle in broader aspect, but they split each phase into more detailed stages later. Detailed stages will be referred to as sub-phases in this discussion and in the thesis. In contrast, Bank's (2000), Law & Kelton's (1991), Shannon's (1998), Maria's (1997) and Balci's (1998) approaches confer detailed breakdown of the project life cycle, consisting of the stages similar to the sub-phases as stated previously. In general terms, all of the documented procedures address the practical issues facing a modeller interacting with the client and developing a simulation model. Although there are inconsistencies in breakdowns and usage of vocabulary, the concept in all approaches remains same. The following key characteristics were found among the procedures described previously.

- The procedures discussed previously indicate that project procedure is not a sequential process and much iteration is involved at various stages and steps in simulation projects.
- The '40-20-40' rule in Shannon's (1998) procedure well illustrates the importance of the planning stage, before the real model-building process commences. This is well supported by Robinson (1994) saying that 'the time spent before developing the computer model, is not the time wasted, but certainly the time saved due to the complex and iterative nature of the simulation projects'.
- According to procedures proposed by Bank (2000) and Law and Kelton (1991), data collection stage needs to commence as soon as the objectives of the study are set. Sadowski and Grabau (2000) recommends such a need by saying that the data collection needs to be commenced at an early stage, as it is considered as the most aggravating, challenging and time-consuming aspect in a simulation project.
- Iteration in validation process in individual procedures, particularly in Balci's procedure (1998), indicates that the validation of outcomes from almost all steps is very important for maintaining the credibility of the model and the project.
- According to Maria (1997), all tasks in a simulation project may not be possible or may not be required for each project and tasks depend on the nature of the project.

The review of different procedures raises the question 'which procedure should be adopted for documentation?' The answer is not straightforward as such an investigation may increase the scope and the time frame of the research. Hence, a revised project procedure would be considered for documentation and discussed in Chapter 03.

2.3 Computer Simulation Tools

Computer simulation is used for studying a model of a real system by numerical evaluation using software designed to imitate the system's features (Kelton et al., 1998). Therefore, for the representation of a simulation model on a computer, some form of programming is necessary. Any means that can assist in the development of the simulation model is called a simulation tool (Oakshott, 1997). A numbers of different forms of simulation tools are available for modellers today to develop the required model. According to Law and Kelton (1991), any simulation tool should offer the following functions for representing the model in a computer, regardless of the modelling approaches – event, activity, process-based:

- Generating random numbers,
- Advancing simulation time,
- Determining the next event,
- Adding and/or deleting records,
- Collecting and analysing data,
- Reporting the results, and
- Detecting error condition.

During the early days, many simulation models with the above steps were created using high-level languages like FORTRAN, Pascal, etc. However, due to complexity of the programming task with the general-purpose programming languages, a wide range of simulation tools have become commercially available. They provide most of the features in a model with less or no programming effort and requiring less experience of the user.

2.3.1 Classification of Simulation Tools

Different authors have their own ways of discussing the various types of simulation tools available. Their views are based on either the extent of the programming task involved or degree of interaction making with the end user in respect of the modelling process.

Law and Kelton (1991) divide the software into two main categories, those which require programming languages and those that allows the modellers to model the system contained in a specific class of systems with little or no programming. The former category includes general purpose programming languages and dedicated simulation languages. The latter is referred to as simulators. Based on the same principles, Shannon (1998) suggests that the modellers have three generic choices in formulating the computer model, namely, general-purpose languages, general-purpose simulation languages, and special purpose simulation packages (simulators). Although the general-purpose programming languages can be used, they are used very seldom today. This is further supported by Bank (1992) and Robinson (1994) dividing software into simulation languages and simulators.

Proceeding in the same direction, Eldabi and Paul (1997) classify simulation software beyond the simulators. They name the simulators as data-driven simulators and add 'programme generator' category to simulation software, enabling more non-computer specialists to produce programme source code from systems description. CAPS, ECSL, VS7 and Draft are considered as examples for programme generators, but they sum up by highlighting that the features of programme generators lie between those of simulation languages and data driven simulators.

A rather richer classification is found in Pidd (1992). Accordingly, the simulation tools fall into seven categories starting from do-it-yourself (general-purpose languages) to Visual Interactive Modelling Systems (VIMS) like Witness, SimFactory, Xcell+, etc. through flow or block diagram systems such as Hocus and GPSS. Instead of this lengthy categorization, the same author (Pidd, 1996) divides the software simply into three categories. The software that requires all users to develop some true programme with, or without, support for visualization comes under the first category. VIMS are the second category, which are primarily based around some kind of visualization for virtually all functions. This includes Witness, Microsaint, ProModel, AutoMode, etc. that comes under the simulators or data-driven simulators, as classified by the previous authors mentioned above. The third category, 'layered systems', is a combination of the first and second categories.

2.3.2 Analysis of Simulation Tools

The review conducted on simulation software revealed that any computer modelling process uses one of the modelling approaches – event, activity or process-based. Internally, those approaches in model building are similar, incorporating a simulation clock, an even list, a timing routing, etc. They differ mainly in the construction facilities available for the user to model a system (Pidd, 1992). Regardless of the approach, the simulation models are built either with broad choices, namely, a traditional general-purpose programming language or a simulation software packages.

2.3.2.1 Traditional General-Purpose Programming Languages

Traditional general-purpose programming languages, such as FORTRAN, Pascal, Visual Basic, and C are rarely used nowadays due to the complexity of the simulation problem and the high requirement of programming skills. However, they are not totally ignored by some modellers because they are flexible, available at a lower cost, familiar with syntax, efficient in execution and applicable to many purposes. All of them are used in procedural style in programming.

2.3.2.2 Simulation Software Packages

Although simulation models can be built using general purpose programming languages, most simulation studies today are implemented using simulation packages. The major advantages of simulation packages are a reduced programming requirement with the introduction of menu driven capabilities; and a natural framework for simulation modelling, as they are more or less specific to particular application. Hundreds of PC Windows-based simulation packages are available in the market today, for different applications, from various vendors and at different prices (Maria, 1997) for the convenience of modellers. However, the review conducted on simulation software packages revealed that they fall into the following categories. They depend on the extent of the programming task involved or degree of interaction made with the end user with respect to the modelling process.

- **Simulation Languages:** They are general in nature, but they may have special features for specific applications like manufacturing, etc. A model is developed by writing a programme, with the pre-constructed programmes, with a little effort, but it still requires programming skills. GPSS and SIMAN are the most common examples.
- **Visual Interactive Modelling System (VIMS):** They are mostly domain-specific simulators as classified by Law & Kelton (1991), Banks (1996), Shannon (1998) and Robinson (1994) or data-driven simulators as classified by Eldabi and Paul (1997), consisting of visual interaction with virtually all of the modelling process, through the Graphical User Interface (GUI). This may also be considered as a hybrid system having commonalities with both flow diagram systems and the Visual Interactive System (VIS) as described by Pidd (1992). Witness and Xcell+ are good examples of such VIM systems. They do not intend the user to be fully conversant with the internal operation of simulations or to be involved with any programming. However, according to Law and Kelton (1991) and Pidd (1992), such systems are limited to modelling only those configuration allowed by their standard features. The adoption of 'programming-like' facility to such systems offers more capabilities for modelling.
- **Layered System:** This type of system can be considered a **hybrid system** of VIM and the general-purpose languages or simulation languages, where the user has both menu driven and programming options. Pidd (1996) comments that with this type of system, the user can operate at a number of different levels such as visual interactive modelling, automatic programme generation, direct coding and a low level of 'bit twiddling'. Arena and Quest (Oscarsson and Moris, 2002) are considered as two of such currently available systems, where both visual interactive modelling as well as programming facilities are incorporated with VIM interface and/or separately.

2.3.3 Different Styles in Simulation Software Packages

In general, early software packages were written in either assembly languages or general-purpose programming languages with a procedural nature. However, the recent trend in software packages is towards the object-oriented programming language (Oakshott, 1997).

2.3.3.1 Traditional Paradigm in Simulation

Traditional simulation modelling has been performed using procedural languages such as FORTRAN, and C (Pratt et al., 1994). The logic in procedural languages is implemented in a usually long sequence of statements, which require both initiative as well as arduous work to function correctly (Chu, 1997). Many simulation languages maintain the same approach, for example GPSS, SLAM, and SIMAN (Joines and Roberts, 1999). They, in general, are the simulation versions of the 'library' approach to simulation and consist of a collection of functions and procedures that are accessed by high-level constructs. Although the use of such high-level constructs allows a more natural modelling of a system than traditional general-purpose languages, they are still based on the procedural style, which corresponds to methods and algorithm of the package instead of real world components. This situation will cause a fundamental difficulty in communication between the simulation code, provided by vendor, and the user code from general programming languages that are required to cope with a complicated modelling situation.

Moreover, Joines and Roberts (1998) comment that lack of extensibility is a major limitation in procedural style, where the changes in the model can only be accommodated by changing the procedure. In simplified terms, the user could add the structural functionality to the simulation but pre-defined functionality of basic processes cannot be altered, except by the vendor. Although some languages available nowadays allow an opportunity for programming the procedure to a certain extent, none of them is fully satisfactory (Joines and Roberts, 1996). As highlighted by Pratt et al. (1994), this traditional approach severely limits the reusability of the model due to lack of extensibility.

2.3.3.2 Object-Oriented (OO) Paradigm in Simulation

Recent years have seen a substantial growth of interest in the subject of object-orientation (OO) and its application to simulation. This is because OO concept focuses on the representation and manipulation the real world as being composed of objects (for example, parts, workers as well as part routing, schedules in a manufacturing cell). These objects are holistic units, not only containing information, but also having the capability to perform different tasks interacting with each other. In the procedural approach, both data and instructions are embedded together. In contrast, in the OO approach, the sequence of computer instructions of the objects (code or function) and the information, on which the

instructions operate (structures) objects, are kept apart. This permits, not only natural representation of the complex real-world system (Chu, 1997), but also produces modular and reusable models (Chatzigeorgiou and Stephannides, 2002). As a result, the developer may achieve a better and faster transition of the conceptual model into a computer model.

2.3.3.3 OO Environment on Traditional Paradigm

Several attempts have been made to use the OO environment to enhance the traditional techniques. Shi (2000) examines the possibility of integration of OO with activity-based construction (ABC). The author describes the activity as containing six classes of attributes, describing the characteristics of the activity including duration, logical sequence, resource requirement, etc. The author comments that accommodation of these attributes enhances the capabilities of ABC. Hung and Iyer (1998) present a similar approach, where a compiler is implemented to convert a process-oriented model to an event-oriented model, which provides a minimum run-time system overhead. For this purpose, a process-oriented model has been constructed with an object-oriented hierarchical framework. Implementation of this framework has speed up the process-oriented simulation. Such attempts prove that OO paradigm can be applied for traditional approaches to enhance the modelling capabilities.

2.3.4 Integration Interfaces in Simulation Software Packages

Nearly all the simulation software packages provide integration interface with other software for easy data transfer. For example, Arena provides ODBC, OLE, VBA and DXF interfaces; ProModel offers OLE interface. Communication between the simulation software packages and the other software is generally accomplished through Dynamic Data Exchange (DDE), which most of the simulation software facilitate. One of such an approach was accomplished between Taylor II and MS Excel by Anderson and Olsson (1998). Availability or establishment of such integration interfaces accelerate the modelling within a more convenience environment. The technique proposed by Seppanen (2000), as illustrated in Fig. 2.6, is another approach, in which the author designs the Excel workbook to strength simulation modelling process in Arena through VBA.

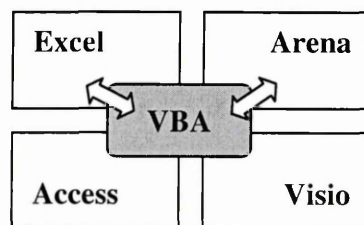


Fig. 2.6 : Potential VBA Data Exchange (Seppanan, 2000)

From model documentation point of view, the availability or establishment of such a data exchange capability may be useful for developing an integrated documentation system to record project details concurrently with the model development process.

2.4 Reuse in Simulation

Reuse generally means that previously acquired concepts, knowledge, etc. are used again and again in future applications, with, or without, changes (Kovacs, et al., 1999). Simulation software and simulation models can be regarded as candidates in this respect (Pidd, 2002). Indeed, the latter is a product of the former. Reusability of software has become a well-established requirement in the software engineering community in the last decade and its importance in the field of simulation has also stressed by many authors (Paul and Taylor, 2002; Kovacs, et al., 1999 and Pratt, et al., 1994). The review conducted so far revealed that there are many different approaches to achieving the reusability in software engineering. The use of the same concepts for simulation models would be useful to improve the productivity of modelling process with model reuse. There is no doubt that simulation modelling primarily concerns itself with ongoing use and alteration of models. Therefore, taking advantage of existing concepts and knowledge would reduce the amount of additional effort and time (Kwon and Park, 1996). However, the spectrum of the model reusability has been addressed by different authors from different viewpoints. Hence, they need to be reviewed to identify the effect of model reusability on documentation, particularly with model representation.

2.4.1 Reuse Spectrum

Pidd (2002) has identified four different types of software reuse with respect to modelling in simulation. They are code scavenging (usage of codes), function reuse (use of built-in functions), component-reuse (module reuse – more than function reuse), and full model reuse. In this approach, he has attempted to describe the frequency and complexity of usage of each category. Generally, the frequency of usage goes from lower to higher, from full model to code scavenging, whilst the complexity acts in the opposite direction. From the simulation modelling point of view, it is true that the first three types are facilitated by simulation software whilst the last category solely depends on the modelling approach.

Proceeding in the same direction, but simply, Paul and Taylor (2002) identify three opportunities to save the time and effort on building the model by reuse, namely, reuse of basic components or elements, reuse of subsystem models and reuse of a similar model. However, they argue that the software package-dependent component makes the reusability ineffective, not only in different domains, but also in the same domain. This highlights a need for software independent reusable components for modelling. In respect of the second category, they conclude that for most cases (except for subsystems having simple components), the reuse of a subsystem model could be more costly than developing it from scratch. This is because that the modellers have to spend a great deal of time understanding how the subsystem components work. Similar argument has taken place for reuse of similar

models. In other words, the authors declare that the model reuse is dependent on being able to trust the reliability of the original model. If a modeller cannot trust a model then surely they cannot reuse.

The evidence suggests that the reliability on model reusability can be built by designing generic and reusable components. Daum and Sargent (1999) also emphasized a need for reusable model elements for effective model development and reuse. Documentation of details of sub-models and whole models, which are constructed with reusable model elements, may build up reliability on reusing sub-model or complete models.

2.4.2 Components/Elements Reuse in Simulation

Many authors state that the availability of reusable components at different levels of abstraction, along with development phases in both software engineering (Castano and Antonellis, 1997; Thomas, 1997; Kwon and Park, 1996) and simulation (Pidd et al., 1999, Chen and Szymanski, 2001 & 2002; Son et al., 2000; Mertins, et al., 2000; Daum and Sargent, 1999), provides a valuable support to the design and implementation of the project by improving simplicity, productivity and quality. However, the reuse of components is just not happening; rather, components must be designed for reuse in the simulation software development stage, enabling the modelling effort to take advantage of reusable artefacts (Thomas, 1997). This is generally achieved in simulation software packages through model libraries or templates or some other means interfaced with software (Pidd, 2002).

2.4.2.1 Reuse Components in Simulation Software

According to Chen and Szymanski (2001), a good simulation software package should have two essential features. They should support reusable modules and the model should be easy to build from scratch. Then, they declare that most freely available software follows a bottom-up approach, making model building straightforward, but limited reusability. On the other hand, most existing commercial software provides a reusable model components library, often come with a friendly graphical user interface. However, they make the model development difficult, with varying degree of complexity in model building and the fidelity of visualization. These problems have been addressed by different authors and they have attempted to develop different techniques to build the components in reusable forms.

2.4.2.2 Building Reusable Components

Among many, the template library approach presented by Mertins et al. (2000) can be seen as a building block of a pre-defined component of a manufacturing system in a simulation scenario. Reuse of simulation models for different scenarios is a major advantage of this method. However, a need in hard programming to integrate different simulation model is a major disadvantage, as it eradicates the easy programming that any software is to facilitate.

Son et al. (2000) presented a more neutral method, which is also based on the development of libraries of simulation components and model templates. The components consist of detailed, formal, information models of all commonly used simulation components such as queues, machines, transporters, etc. These components are tailored to specific modelling scenarios, which can be defined by different modelling templates, as illustrated in Fig. 2.7. Model builder generates a model for specific commercial software from these neutral descriptions of the components. The availability of such libraries, together with respective translators, would simplify and accelerate the model building process. They declare that such an attempt enables, not only model reuse with reusable components, but also speeds up internet-based simulation service. It seems that this structure that consists of header information, experimental information, shop floor information, product/project information, production information and output information, has supported model documentation to some extent, though the users have been unspecific about such benefits. The methodology was designed and implemented for Arena and ProModel software packages. However, they have not addressed how the component reuse in generic form (they suggest different templates for each different domain) is achieved. Therefore, this approach for a generic software independent structure would be questionable.

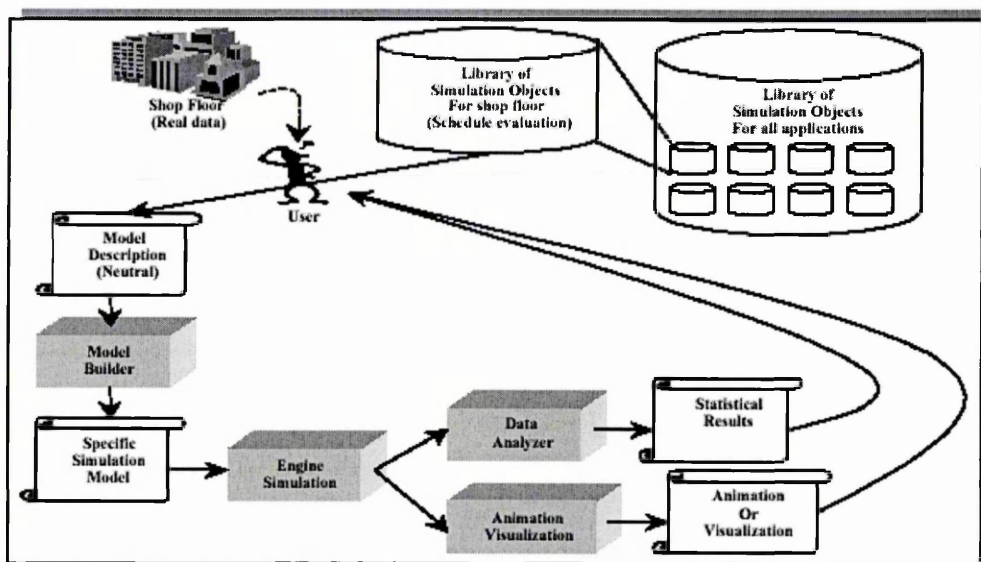


Fig. 2.7 : Neutral Library Concept to Develop Simulation Model (Son et al., 2000)

Having revealed many advantages in component reuse in respect of developing large scale models, building models in reusable form, linking models in distributed operation, accelerating web-based operation and providing directory service to application developers (they imply the future), Pidd et al. (1999) and Buss (2000) declare that the component based work in simulation is primarily based on or complementary to the OO paradigm. Component work is known to be feasible and underlies current commercial simulation software outside the domain of discrete simulation. But, with the evident in approaches such

as HLA and COBRA, it is possible to take existing computer programs, which are not object-oriented, to wrap up them inside an object-oriented shell. Leaving how the OO components are unwrapped into individual software procedural element for the moment, an attempt made by Pratt et al. (1994) to build up reusable modelling framework is briefed, as the concept would be useful in developing a generic framework for documentation.

2.4.2.3 Separation of Model Elements for Model Reusability (Bhuskute et al., 1992;

Duse et al., 1993; Pratt et al., 1994; Delen et al., 1992; Delen et al., 1996)

A group of researchers at Oklahoma State University attempted to present a conceptual framework for simulation modelling that emphasizes the model reusability. In their attempt, reusability is addressed through the separation of model components or elements. The framework is a tribute to the modular library database and it can be implemented within the traditional modelling environment (procedural based) and more superiorly in the OO environment. Then the tool-independent base model was constructed through the separated elements so that the base model could be translated into any other executable model. This description illustrates that the authors have attempted to address the important factors, such as generic, reusability, modularity, tool independent, and so forth. These are the important characteristics in model documentation framework also and thus the approach is dealt with in detail.

➤ Criticism in Traditional Simulation Modelling

Pratt et al., (1994) begins the discussion about the separation concept by presenting the problem encountered in the traditional paradigm in simulation, with respect to the reusability. The following are the significant phrases or problems stated in the traditional approach that are found in most existing commercial software.

- Once the objective of the project is realized, the model is often discarded (i.e. single objective model or throwaway model) though some elements contained in the model could be reused for a new model.
- Lack of access to models by non-modelling specialists limits their usage.
- As the traditional components perform more than one function (one-to-many-functions), their usage is limited to situations for which it is designed.
- The details in the control and information aspects of the model are hard coded and spread through out the model. Hence, the model is hard to modify and to use for multiple purposes.

The above, all emphasize the difficulties in altering the model itself and in reuse of the existing elements. To overcome these problems, Bhuskute et al., (1992) and Pratt et al., (1994) present the concept of separation for abstraction of real-world entity.

➤ Separation of Model Elements

According to the separation concept, any entity in the real world has three dimensions, viz. physical, information and control or decision. The modeller can separately focus on these three dimensions for representing the real world as modelling objects. Therefore, abstraction automatically results in separation of the above facets of an object. Despite having these three dimensions, the object can still be classified as one dimensional, depending on the primary focus. This emphasizes that elements or components of the real-world system can be represented with physical, information (information flow) and control/decision elements.

- **Physical Element:** A physical element is an object with a tangible correspondent in a real-world setting. In other words, if the modellers' primary focus on the object is its physical characteristics, then the object is considered as a physical element. Visualizing physical elements is straightforward even for non-specialists. Examples in this category include parts, machines, buffers, material handlers etc. in a manufacturing system.
- **Information Element:** An information element may, or may not, have a real tangible correspondent in the real-world system like physical objects. If the modellers' primary focus on the object is its information characteristics, then the object is designated as an information element. Grasping information elements is not as easy as physical elements, but a little exposure to the simulation environment makes it easier. Examples include operations, part routs, etc. in a manufacturing system.
- **Control/Decision Element:** A control/decision element is a logical object, which typically has no tangible correspondent in the real-world system. In that sense, control elements are potentially more difficult to grasp. An object can be classified as a control/decision object, if its primary function is to:
 - Control and evaluate the state of a given system,
 - Exercise a logic algorithm, and
 - Signal an appropriate action to be taken.

Examples of control elements include queue controller (priorities), workstation controller (job loading), etc.

The separation concept is primarily based on the OO paradigm, which seeks to tightly couple an object's data and the procedures for manipulating the data. However, the OO paradigm is tightly coupled with a computer science issue. It primarily encounters the computer's internal mechanism for manipulation of data and access to data structures and procedures, used in creating useful programs and applications. In contrast, the separation

concept is a modelling and user issue. This provides a mechanism through which a modeller can structure the components needed to construct a total simulation model in a reusable form. It indicates that this approach of creating reusable models has little to do with the internal representation whether it is procedural based or object-oriented. This shows the ability to implement the approach for both paradigms, for which this research needs to be involved.

➤ **Benefits of Separation Concept**

The benefits from the reusable model elements can be seen in short-term as well as long-term contexts. In the short-term context, it allows the modeller to think of these elements independently during the model development process. Hence, it permits the modeller to build the model in a more natural model-building environment. For example, the process involves selecting the appropriate physical elements without being constrained by model information flow. This natural approach facilitates creating models with a higher degree of integrity and greater flexibility. In respect of the long-term context, the framework allows the modeller to couple the several primitive elements (or existing primitives with new ones) to form a coupled object that corresponds to real objects. In other word, boundaries of the system and alterations to deal with different objectives could easily be managed within this framework.

➤ **Implementation of Separation Concept**

Regardless of the benefits expected within the time frame, the significant advantage of this framework, which is based on OO, is the ability to implement it within the traditional modelling environment (Pratt et al., 1994) as most existing commercial software packages are procedural based. The applicability of the framework or separation concept for the traditional paradigm is illustrated in Fig. 2.8, which demonstrates the separation concept for information (parts arrival) and decision element (SEL-queue selection rule).

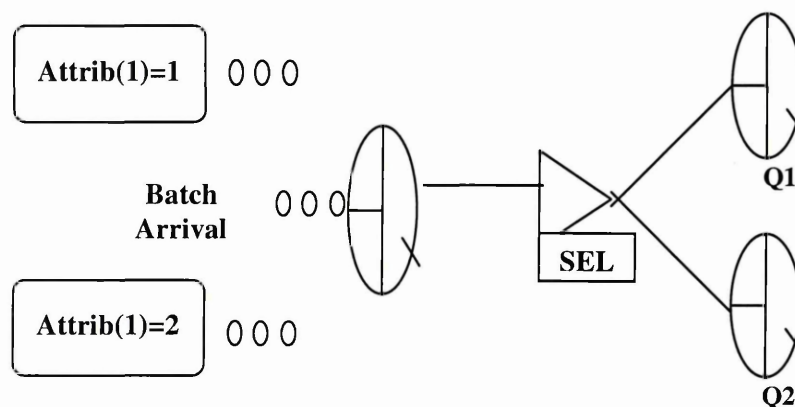


Fig. 2.8 : Separation Concept in Traditional Paradigm (Pratt et al., 1994)

The concept has been implemented on OOM environment by allowing the modeller to extend the general elements (parent) to specific elements (subclasses). It becomes a more natural and easier modelling environment than the traditional approach due to inheritance characteristics in OO. The concept is illustrated by a machine object, in which physical elements (processor and queues) interact with control element (controller) as in Fig. 2.9.

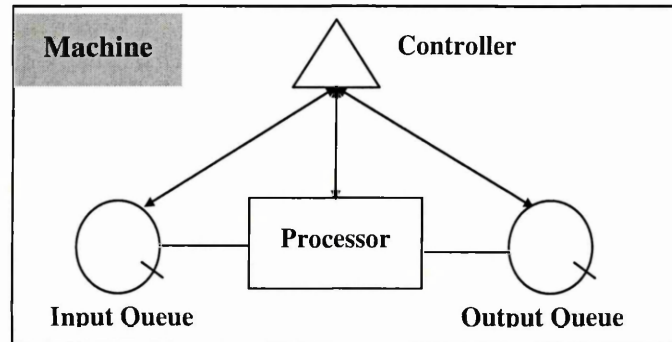


Fig. 2.9 : Primitives in a Machine with OOM Paradigm (Pratt et al., 1994)

The above example shows, not only the implementation of the separation concept, but also a concept for building composite objects. Nevertheless, it raises an issue whether the composite objects are applicable with the traditional paradigm for the benefit of long-term reusability of sub models or a whole model. It seems that in respect of traditional approach, the concept is justifiable for short-term reusability with component/element reuse, but the long-term reusability is questionable. However, as stated previously, components/element reuse is the major issue to be considered in the documentation framework.

➤ Separation Concept for Base Model

The richness of this separation concept is further enriched by Delen et al. (1996) by adopting the concept for tool-independent model representation, in which they refer the model as 'base model'. This can be considered as further development or implementation to the framework suggested by Duse et al. (1993). From border perspective, the Delen et al. (1996) realized reusability of the separation concept is limited to the simulation, for which it is designed. Therefore, an attempt is made to extend the concept to other analysis tools like layout design model, queuing model etc. This is achieved by applying the separation concept to create a base model, which Duse, et al. (1993) is considered as a tool-independent model. According to Duse et al. (1993), within the base model framework, a model of the enterprise should be created without keeping any tool or any specific problem as the context for model development. The modeller can construct the base model by simply selecting appropriate elements and assembling them in a certain software environment. Once the

base model is created, it is then translated (Base-Model-to-Tool-specific-translator) into tool-specific model [Delen et al. (1996) refer to it as execution model] by extracting the required structure and parameters. Tool-Specific-Package-Specific-Translator converts the model into a package-specific model for experimentation.

The area of this research has been carried out involving more than seven completed Ph.D. researches. Prototype software, VisualWorks 2.0, has been developed (Pratt et al., 1994) and it was proven that this framework is applicable for multi-tool modelling (OSU-CIM, 1997). The separation concept delivers tool-independent elements, which are generic, reusable, modular, and even applicable for traditional simulation paradigm. Hence, this concept may provide a context for software independent model documentation.

2.5 Model Representation (MR)

Most of the simulation project procedures explored previously indicates that a conceptual model is typically represented diagrammatically in some way, prior to the analysis model being coded. Such a model is known as a diagrammatic model, a graphical model, or a system description model. This is a step towards the executable or analysis model, which may have already been envisaged with many limitations and assumptions. In such a model, the system features such as entities, resources, processes, rules, etc, are represented by graphical shapes to display the interactions between them. In other word, such models exemplify the fundamental features of static simulation models as imitations of real system structure and their operation. However, a full or part of static representative model may be translated into an executable computer model, depending on the tool being used for model representation (Bortscheller and Saulnier, 1992 and Whitman et al., 1997). Besides this translation, the model representation in general provides many more benefits to the simulation project itself as well as to the simulation community. According to Ceric (1994), Pooley (1991), and Bortscheller and Saulnier (1992), the following are the benefits expected from such a sound representative model:

- It is essentially a form of self-documentation.
- It simplifies the task of understanding the logical design of a model.
- It encourages high-level thinking.
- It facilitates communication tool between developers and other parties.
- It provides a sound foundation for model verification and validation.
- It may lead to the formulation of small and simple models.

Although theoretically, it may sound useful; the benefits depend on the techniques or tools used for the model representation. More than 20 alternatives of representation techniques can be found in literature and thus making the selection of a technique for model representation from the documentation point of view is much more difficult.

2.5.1 Classification of MR Techniques

Nevertheless, the following classification proposed by Ceric (1994) makes the choice a little convergent.

- **Simulation Strategy Neutral Methods:** These methods do not belong to any of the simulation strategies or to any simulation languages. Neutrality of these methods enables their use for any simulation strategy or simulation software. It appears that Flowcharts, Activity Cycle Diagrams (ACD), IDEF techniques, and Petri Nets (PN) are included in this category.
- **Simulation Strategy-Oriented Methods:** These methods inherit the characteristics belonging to a particular strategy, but they are not software or language specific. Event graph, which is based on event-oriented simulation, is a good example of this category.
- **Simulation Software-Oriented Methods:** These methods show the characteristics of particular software or languages. The most common in this group are GPSS transaction block diagrams, SLAM network diagrams, etc.
- **Methods Borrowed from Other Computer Modelling Areas:** Any methods borrowed from computer modelling in order to enhance model representation in simulation fall into this category. Unified Modelling Language (UML), which is treated as the industrial standard for the presentation of the software design model, is considered as this kind of method. However, the methods under this category may be considered as neutral method that is applicable to simulation.

Accordingly, any method in the first and last group is appropriate for software-independent model documentation purpose. Therefore, IDEF, PN, UML, ACD, and flowcharts are considered as potential model representation techniques for documentation. This presents the necessity of comparison of the neutral techniques (including the last category) and for establishing the criteria for comparison process.

2.5.2 Comparison of MR Techniques

Little attempt has so far been made to evaluate the relative merits of all model representation techniques available today, particularly in the context of model documentation. However, a very few attempts have been found in early stage to compare very popular techniques. Therefore, only the potential model representation techniques suggested above (neutral techniques including the last category) are discussed in the view of finding the most appropriate method for model representation and documentation framework.

2.5.2.1 IDEF Tools

IDEF [Integrated computer aided manufacturing (ICAM) DEFinition] is a systems definition method and was developed under the ICAM project of the US Air Force. It aims to describe the information and the organization structure of a complex manufacturing system. The aim is achieved through multiple IDEF tools, extending the family from IDEF0 to IDEF5 (at the current literature) including IDEF1X (Pandya, 1995; Kateel et al., 1996; Pandya et al., 1997; Whitman et al., 1997 and Delen et al., 1998). At a glance, the individual members of the family describe the perspectives: IDEF0 – functional or activity modelling, IDEF1 – Information modelling, IDEF1X – data modelling, IDEF3 – process flow and object state description capture method, IDEF4 - object oriented design capture method, and IDEF5 – Ontology description capture method. IDEF2 (dynamic modelling) is not in the list as it was intended to be used for simulation. SLAM simulation language is the result of graphical notations of IDEF2 (Whitman et al., 1997). The brief review of the above methods shows that the choice for model documentation lies between IDEF0 and IDEF3.

- **IDEF0 (Functional Modelling Method)** is a method designed to model the decisions, actions, and activities of an organization or a system. IDEF0, derived from a well-established graphical language - the Structured Analysis and Design Technique (SADT) (KBSI, 1999)-, is capable of representing a wide variety of business, manufacturing and other types of details graphically. The methodology combines both graphics and texts; and presents them in an organized and systematic way, making it easy to understand and use. Decomposition of the method allows using for both top-down and bottom-up modelling. These are achieved through its basic constructions, which is a function block, linked to other function blocks by inputs, outputs, mechanisms, and controls. Links between the blocks may be physical objects, such as either material flow, or information flow, which is cited as one of the major drawbacks (Pandya et al., 1997). In addition, there is a tendency that the readers may misinterpret the model as an activity sequence, any sequences is even not included. Such a drawback is addressed in IDEF3.
- **IDEF3 (Process Description Capture Method)** provides, in summary, a mechanism for collecting and documenting processes. It captures precedence and causality relations between situations and events in a form natural to domain experts, by providing a structured method for expressing knowledge about how a system, process, or organization works (KBSI, 2000). Specifically, it is capable of:

- Recording the data resulting from interviews in systems analysis activities,
- Documenting the decision procedures affecting the states,
- Determining the impact of an organization information resource on major operations,
- Making system design, and
- Providing simulation model generation.

There are two IDEF3 description modes: process flow and object state transition. A process flow description captures knowledge of 'how things work' in an organization. The object state transition network (OSTN) description summarizes the allowable transitions an object that may undergo throughout a particular process. The basic elements of the process diagram consist of processes, which can represent operations, decisions, procedures, scenarios etc., links, which represent the relationship between elements, and junctions, which show the convergence or divergence of multiple process flows and their timing. OSTN diagrams capture object centred views of processes, which cut across the process diagrams and summarize the allowable transitions through object status and referents.

Both IDEF0 and IDEF3 provide benefits towards to documentation in respect of neutrality, descriptivism, recognition, and generality. However, Oscarsson and Moris (2002) compare IDEF0 with flowcharts and recognize that flowcharts are user-friendlier than IDEF. Similarly, the survey conducted by Schormann and Perera (1999) on model representation techniques shows that IDEF (0&3) and flowcharts have similar overall features (ranked as same) according to their criteria of the assessment.

2.5.2.2 Activity Cycle Diagrams (ACDs)

ACDs are particularly useful for systems with a strong queuing structure (Paul, 1993). It describes a problem space with entities and the states the entities at any given time. The states can typically be: dead state [the entity is idle or in a queue (either a real queue in the problem space or a notional one defined to model a delay in the problem space)] or live state (the entity is active or is engaged with other entities in some time taking activity). Two simple symbols: a circle and a rectangle with arrow represent the states. Therefore, use of these two symbols is the major advantage in this method.

Although ACDs are powerful and simple to use, the approach has certain limitations. Simplified formulation of a potential complex problem may cause difficulties as ACDs do not provide the modeller with the necessary representations to capture their problem adequately (Pidd, 1992) and do not allow presenting the attributes of entities. These limitations in ACDs may have led to the development of other forms of ACDs (Odhabi et

al., 1998). Consequently, Extended ACDs (X-ACDs) (Pooley and Hughes, 1991) was proposed and then it was further developed to Hierarchy ACDs (H-ACDs) (Kienbaum and Paul, 1994). Similarly, many other developments like SH-ACD, ML-ACD, etc can be found in literature. Such a development makes the choice of ACD for model documentation purposes much more difficult. Inappropriateness of ACD for model representation is further confirmed by the lowest overall rank taken (among IDEF, ACD & flowcharts) in a survey conducted by Schormann and Perera (1999).

2.5.2.3 Petri Net (PN)

PN is a graphical and mathematical modelling tool that can be used to perform both static and dynamic analysis of processes of a system (Sawhney et al., 1999 and McLean et al., 2002). It is used as a visual-communication aid similar to flowcharts, block diagrams, and networks. It consists of four basic elements: place, transition, arc, and token. A place, denoted by a circle, represents a condition such as input data, input signal, resource, condition, or buffer. A transition, represented by a solid bar, displays an event such as computation, task, or activity. Transitions are active components, which can fire and change the state of the system. Fire is allowed only if transitions are enabled. Arcs, depicted by arrows, are used to connect places and transitions. They are either drawn from a place to a transition or from a transition to a place. Arcs can also have a multiplicity and such multiplicity is represented by an integer 'k' that dictates the number of tokens required to fire or enable a transition. The token element, denoted by solid circle (black dot), provides the dynamic simulation capability. These elements together with PN structure provide both static and dynamic analysis capability. Hence, PN can be considered as a sound candidate for model representation for documentation.

However, there are various kinds of PNs and computer tools for using them. They differ quite a lot, as to their expressive power, legibility of models or analytical capabilities. Some such developments are stochastic PN (SPN), Coloured PN (CPN), Hierarchical CPN (HCPN), Objective Oriented PN (OOPN) and many more (Vojnar, 1997). The development itself displays the power of PN for analysis systems. On the other hand, such enhancement reduces the competence of the document writer and reader. Hence, the appropriateness of PN for documentation is questionable.

2.5.2.4 Unified Modelling Language (UML)

UML is defined as modelling language for specifying, visualizing, constructing and documenting the artefacts of a system-intensive process. It can be considered as a general purpose, broadly applicable and industry-standardized language (Pillana and Fahringer, 2002 and Kim et al., 2003). One of major application areas of UML is the documentation of a

software design framework for object-oriented simulation (Rossetti et al., 2000). As the designing and building a simulation model is often associated with software engineering, it is worth dealing with the applicability of UML for simulation model documentation. Richter and Marz (2000) have made such an attempt with object oriented modelling. UML has a somewhat complicated structure to understand at first glance. In general, there are three main modelling viewpoints, namely, use case models, static models, and dynamic models. Use case models describe the system requirement from viewpoint of the user, by cases and actors. Static models are essentially class diagrams that describe system elements and their relationships, including generalization. Dynamic models describe the system behaviour over time, by state diagrams, sequence, collaboration, and activity diagrams.

In all, it is clear that this relatively new notation is difficult to embrace without much practice. To support this viewpoint, Oscarsson and Moris (2002) highlighted that the use of UML would probably not add any value to documentation. They stated, "Perhaps the UML model will be more difficult to understand than the simulation model".

2.5.2.5 Flowcharts

A flowchart is a visual tool for describing a process graphically being studied or for planning the stages of a project. Therefore, it is useful in an initial process analysis. Steps in a process are shown with symbolic shapes in sequence. The flow of the process is indicated with arrows, connecting the symbols so that the reader can examine the order presented and has a common understanding of how the process operates. In other words, it helps the user to see whether the steps of a process are logical, uncover problems or miscommunications; to define the boundaries of a process; and to develop a common base of knowledge about a process, resulting in powerful, rapid, and clear communication. A good flowchart of a bad process will show how illogical or wasteful some of the steps or branches are (PathMaker, 2002). Computer programmers popularized flowcharts in the 1960s, using them to map the logic of programs. Since then, it has become popular in many areas. For example, in quality control, flowcharts are particularly useful for displaying how a process currently functions or could ideally function (Laudon, 1978).

The use of standardized symbols in flow charts provides a common language to visualize problems and makes flowcharts easier to read and understand. In fact, flowcharts have been used for so long that no one individual is specified as the 'father' of the flowchart. The reason for this is obvious, as flowcharts can be customised to fit into any need or purpose. For this reason, flowcharts are recognized as a very simple, popular, and accepted communication tool in various applications.

➤ Types of Flowcharts

There are three main types of flowcharts that work for almost all situations. They can be used to present details in different levels for different readers of a single application.

- **Basic or High-level Flowcharts:** This type of flowchart maps only the major steps in a process for an overview of the process. Hence, it may be useful for tasks like team discussion. It appears that such a representation is important for documentation, as the model developers do not necessarily to write the model code. Instead, they need to use predefined functions and procedures provided with the simulation software. The Fig. 2.10 illustrates a usage of this type of flowchart.

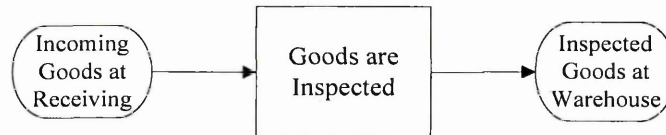


Fig. 2.10 : An Example for Basic or High-Level Flowcharts

- **Process or Detailed Flowcharts:** Detailed flowcharts show a step-by-step mapping of all the events and decisions, which examine the processes in the project. A programme flowchart, which describes the processes within a computer programme in a computer system and the sequence in which they must be executed, is one of the applications of such detailed flowcharts. Although most of the existing software packages provide predefined functions to avoid such detailed representation, from a model documentation standing point, it may still be necessary to represent some aspects of user logics. The Fig. 2.11 illustrates the usage of this type of flowchart for the example shown in Fig. 2.10.

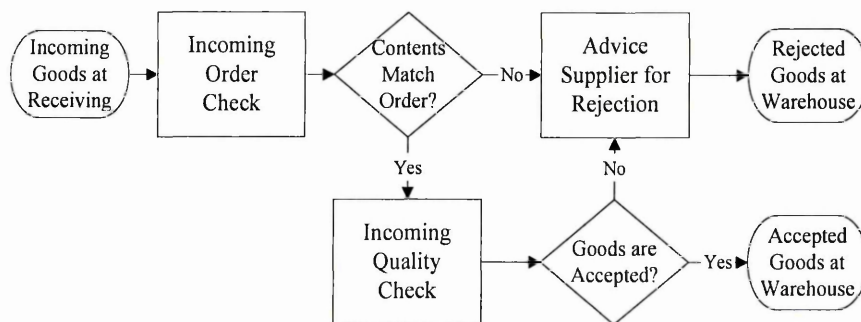


Fig. 2.11 : An Example for Detailed or Process Flowcharts

- **Deployment or Cross-Functional Flowcharts:** Deployment or cross-functional flowcharts are somewhat similar to detailed or process flow charts. They indicate the relationships between process steps and functional units by columns with each column representing a person or department involved in the process. In other words, they are used to capture the flow of a process from department to department or show how a process influences different functional units of a

system. From documentation standing point, this type of flowchart may be important in representing the structure of a distributed simulation model. In general, a vertical layout places slightly more emphasis on the functional units while a horizontal layout emphasizes the process. Fig. 2.12 illustrates the application of this for the same example illustrated in Figs. 2.10 and 2.11.

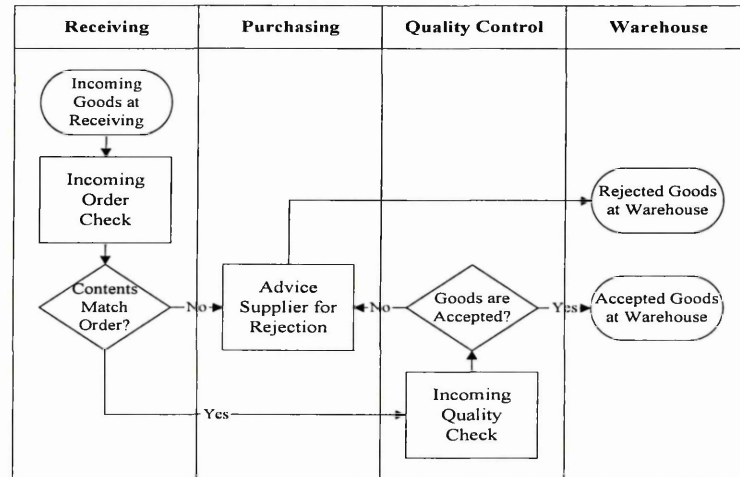


Fig. 2.12 : An Example for Deployment/Cross-Functional Flowcharts

➤ Advantages and Disadvantages of Flowcharts

The pictorial attributes of flowcharts mainly facilitate communication among people. Regardless of the background of people, they tend to understand the symbols and can easily visualize the main features of the process, as the flowcharts use extremely neutral and generic notations. This will be one of its major advantages, when it is used to describe the simulation model. According to Oscarsson and Morris (2002), flowcharts enable people with different background to contribute to the overall discussion. Another major advantage is that flowcharts can easily be customised to suit different situations. The customised feature, with some added facilities, enhances the capabilities of flowcharts to be applicable in diverse environments. It makes the flowcharts more popular.

However, even if it is a widely used technique, Oscarsson and Morris (2002) consider it as often abused. The lack of use of formal procedures sometimes makes interpretation ambiguous or misleading. When flowcharts are too large, they become more complicated to track and understand due to lack of decomposition capabilities.

2.5.3 Flowchart for Model Representation and Documentation

The review on MR techniques shows that there is no perfect method for MR in respect of model documentation. Each potential MR candidates has their own positive or negative capabilities in different perspectives. This may be the reason that the survey conducted by Oscarsson and Morris (2002) recommends following choices for model documentation:

- Use flowcharts and comment on the code for low-level documentation.
- Use UML diagrams for more precise documentation with the description of structure and behaviour of the system.
- Use IDEF0 to describe the system flow in conceptual model documentation.
- Combine IDEF0 and flowcharts to present the documentation of the physical structure of the system.

However, the authors then compare the three MR techniques (Flowcharts, IDEF0, and UML) against the criteria: neutral notation, generic notation, a recognized notation, user friendliness, and descriptive details in several levels. The major differences among the techniques appear in respect of user-friendliness and level of description. Flowcharts are the user-friendliest technique among the others, but it does not show the same magnitude of descriptive power that the others have. This evaluation has resulted in recommending flowcharts for documentation of simulation models.

A similar survey conducted by Schormann and Perera (1999) also shows that IDEF and flowcharts have a similar rank whilst ACD has the lowest rank, according to their criteria: coverage, ease of use, ease of change, clarity and readability, hierarchical diagrams, popularity, range of constructs, reusability, and convertibility to other MR techniques. The survey also has confirmed that simulation practitioners very rarely use any formal model representation method. If any is practiced, then they use simple flowcharts, supported by the symbols provided by the software, which they intend to use. This itself shows the use of flowchart in simulation due to its simplicity.

The review on MR techniques in respect of model documentation shows that flowchart can be considered as the most appropriate method due to its generic, neutral, recognized, and user-friendly features, as the documentation needs to be benefited for different types of audiences from expert to non-expert. However, the poor quality and imprecision of flowchart diagrams, in case of a large complex model, semantic problems and less descriptiveness of details in different levels still have to be tolerated by the user.

2.6 Documentation in Simulation

Documentation in simulation is referred to in different contexts by different authors, but mostly and mainly, model documentation is considered as model representation. Many benefits expected from documentation can be found among the literature, but not much attempt has been made for documenting the entire simulation project. Therefore, the literature is reviewed to understand the documentation function in simulation, to be aware of the benefits from documentation and to identify the documentation capabilities, strength, and weaknesses in current practices, in existing methodologies and in software packages.

2.6.1 Different Perspectives for Documentation in Simulation

Study of the simulation project or project life cycle is the best way of understanding the documentation function in simulation. The procedures presented in Section 2.2 show that documentation is dealt with in different stages with different perspectives. Therefore, the following section summarises the function of documentation presented by different authors.

- Robinson (1994) identifies the following various documents in simulation project.
 - Project specification (introduction to the problem; expected benefits; scope and level of details; assumptions; experimental factors; reports to be presented).
 - Model documentation (a list of elements and variables and their purposes; a summary of model data; an explanation on the model logics – representation).
 - User (i.e. model executor) documentation (objectives; overview of the model; experimental factors, how to change experimental factors; how to run the model and perform experiments; results, how to access, analyse and interpret them),
 - Final reports (executive summary; objectives; key results; conclusions and recommendations).
 - Minutes of project team meetings (decisions and progress made during the project).
 - Implementation project specifications (recommendation to be implemented; who, how and when the recommendations are to be implemented).
 - Project review (further improvements to model; recommendation to further studies).

In this description, the author discusses the possible details to be documented and shows the importance of documentation throughout the project. He has also given an indication on different documents for different audiences.

- Sargent (2000) identifies that both detailed (tests, evaluation made, data, results, etc.) and summary (table for data validity, conceptual model validity, computer model verification and operational validity) documentation are critical for convincing users, the correctness of a model and model results.
- In Maria's procedure (1997), model documentation is placed just after validating the model. In this context, the objectives, assumptions, and detailed input variables are considered as the details to be documented.
- No other author has discussed the details to be documented in detail though they have highlighted the importance of documentation. For example, in Bank's (2000), Law and Kelton's (1991) and Shannon's (1998) procedures, the documentation function takes place at the latter part of the project (i.e. after the experimentation). They stress the importance of reporting results to the client. They also specifically emphasised the necessity of documenting the model, i.e. representative model for future use.

In all, it is clear that documentation is an important, essential activity in a simulation project and it is given a different interpretation, considering the documentation of the model (i.e. model representation) as a vital element, but spreading it throughout the project. However, such varying perspectives bring the following key issues to be addressed in this research:

- **What is meant by documentation in simulation?**
- **Why is the documentation in simulation necessary?**
- **What are the details that are to be documented?**

The above questions are the major issues to be addressed in this research and are dealt with in subsequent chapters. More specifically, the first two questions are answered in Chapter 03, in establishing the documentation framework. The answer to the last question is provided in Chapter 03 and 04 according to the proposed approach for documentation.

2.6.2 Benefits from Documentation in Simulation

Different authors describe the different benefits from documentation within their context of documentation in simulation. Regardless of the context, the benefits can broadly be seen in two perspectives: short-term benefits and long-term benefits. Short-term benefits are the immediate benefits available for the direct and indirect stakeholders such as the client, project team and other in-house members in the project. Long-term benefits are encountered when the model is intended to be reused for some other objectives or to study the model for general interest. However, in both ways, benefits are generally centred around communication, understanding, modification, or alterations. In respect of general benefits (regardless of short-term or long-term), several authors, such as Robinson (1994), Shannon (1992), Banks (2000); Sargent (2000); Benz et al. (1997); Hoch et al. (1998); Musselman (1994); and Bortscheller and Saulnier (1992), state the potential benefits from documentation in simulation. Some of the major advantages in documentation in simulation are listed below:

- It provides depth understanding of the model and the project.
- It acts as a communication tool within the project team and those involve indirectly with the project.
- It acts as a supporting information system when the project is being reviewed.
- It transfers the knowledge between team members and other interested parties.
- It keeps the enthusiasm of the project team high.
- It avoids misunderstanding at early stages of the project, thus minimizing delay.
- It accelerates the development process.
- It convinces the correctness of the model.
- It increases the chances for a successful project.
- It greatly facilitates model modification and enhances the effective model reuse.

- It helps the modeller to learn from previous mistakes and perhaps provides a source of sub-programs or sub-models that can be reused in future projects.
- It ensures that the model can be used even in the event of personnel changes.

In most instances, the above benefits can be expected only if the documentation is performed in parallel with the project development. Therefore, the documentation should be an integral part of the model development process (Hoch et al., 1998). It is clear that the general need for such documentation is visibly declared by the benefits expected from the documentation, whether they are short or long-term. Furthermore, authors (Nance, 1979 and Richter and Marz, 2000) expand their horizon beyond such benefits and general needs, towards a standardized approach for a simulation project through documentation. However, the benefits listed above raise the following key issues to be addressed in this research:

- **Who benefits from documentation in simulation?**
- **What are the details required for individual beneficiaries?**

The first question is a fundamental question that needs to be addressed before the development of a methodology for documentation of simulation projects. Hence, it will be addressed in Chapter 03, in establishing the documentation framework. The second question needs to be answered once the details to be recorded in simulation projects are identified. Therefore, it will be addressed in Chapter 04, after the details are identified.

2.6.3 Current Practice in Documentation in Simulation

Despite many benefits from documentation in short-term (the current project) or long-term (model reuse), some authors such as Keller and Dungan (1999), Benz and Knorrenchild (1996) comment that model details are recorded idiosyncratically as the individual modeller desired. In support of this argument, Maria (1997) stresses that unrecorded assumptions are one of the pitfalls to guard against in a simulation model. However, whenever the documentation (i.e. model representation) is attempted, the common practice is to produce flow charts, which are biased to the simulation software being used, to represent some aspects of model logics (Schormann and Perera, 1999) and to present brief model objectives with project results with varying degrees of details. Or else, it appears that many practitioners opt for direct model coding with chosen software, instead of a lengthy documentation process.

Such an approach is convenient for model authors and speeds up the model building process in the short-term, but it may not represent the entire details, involved with the project. Therefore, it may cause problems in subsequent alterations and analyses that will result 'reinvent the wheel' again and again. In support of this statement, Keller and Dungan (1999) express that such a documentation approach may also not be systematically linked to

each other phase of the simulation project life cycle and to sources of further information. As a result, existing model details would be partially available or ineffective. This would result in an ill-defined final model with varying interpretation. Such problems may rarely be noticed in small models containing a few or several elements, but the serious problems may exist with large and complex models, which may often exist nowadays.

Nevertheless, this raises the key question ‘why don’t the modellers document the simulation projects or why do they do it idiosyncratically as they desired, whenever it is attempted?’ Answer to this would be the major motive in this study and in the development of methodology for documentation of simulation projects. It will be addressed throughout the thesis.

2.6.4 Documentation Approaches in Simulation

As discussed in Section 2.6.1, the documentation in simulation is treated in different perspectives by different authors. In their discussions, some authors consider the MR as documentation or model documentation. Some others treat MR as a critical element in the documentation that includes many other details spreading throughout the entire simulation project. Consequently, the existing documentation approaches can be seen in two perspectives: documentation of the whole project and model representation. With respect to the documentation of whole simulation projects, it appears that no attempt has been made, except a conceptual approach proposed by Nance (1979) through ‘Conical Methodology’. For model documentation, a few direct or indirect attempts can be found in literature and one of such direct attempt is made by Richter and Marz (2000).

2.6.4.1 Conical Methodology for Documentation (Nance, 1979)

Nance (1979) begins the discussion by dividing the documentation into two types: program documentation and model documentation. The programme documentation, similar to software engineering, is on a single level. It targets a user who is fully knowledgeable of the syntax and semantics of the programming language. Hence, the usage of programme documentation is less important. Whereas, the author argues that model documentation must function more than one level for following potential users:

- The top-level manager who funds the modelling effort,
- The top-level manager who consider the application of the model to a problem in a completely different context,
- The systems analyst who is quite familiar with the simulation techniques but unfamiliar with the specifics of the language or simulator used,
- The application programmer who must consider the detailed differences in implementation necessary for translating the model.

2.6.4.2 Model Documentation with UML (Richter and Marz, 2000)

With this approach, an attempt has been made to apply UML to design and build the simulation model. Because of the features available in UML to build static and dynamic models, the authors express that it provides the common language for communication between the user and developer. The documentation is built parallel to the model development process. In addition, it provides a sound base for identifying reusable components in simulation models. According to their comments, this approach establishes a general standard for modelling and for documentation, but it is limited to the model representation and building aspects in simulation only. Furthermore, as discussed previously under model representation techniques, UML is a relatively new tool and may cause difficulty in grasping the concept by variety of audiences. Therefore, the use of UML for model documentation would be doubtful in the context of this research.

2.6.5 Documentation Capabilities in Existing Simulation Software

No existing software was found to facilitate the documentation of the entire simulation project. However, various facilities, like note-pad, code-comments, etc. are provided in different software. They facilitate recording of helpful notes in respect of the project or, in particular, on model codes and the building process. For example, Witness (Lanner, 2003) facilitates for creation of customized reports on the structure and details of the model and model logic interacting with the model. This is achieved through the menus, such as model notes, flow rules, etc, in Witness Documenter. In AutoMod (Simule, 2001), comments can be added to the procedure code to display a description about the model codes. Models in Arena (Kelton et al., 2002), itself provide a comprehensive description about the model and it has also been enhanced through Visio documentation tool interface.

However, no software has provided a specific structured method to document the model or other details. Nevertheless, as discussed in Section 2.3.4, evidence shows that the documentation can be facilitated by interfaces or any other means provided in different software. Even when such a capability is available, the method of presentation is left to the individual user, producing the documents with vague information. In other words, no software has a built-in standardized support for documentation. Such drawbacks make it difficult to track how a simulation project was carried out, or even how the model was built.

Overall, it appears that there is no structured methodology or any built-in software support for documentation of the entire simulation project, though the need and benefits are emphasized and discussed from different viewpoints and perspectives.

This situation has led to extend the literature survey for documentation of other related areas, such as software development and other modelling arena.

2.7 Documentation in Software Development Process (SDP)

Designing and building a simulation model is often associated with the software development process. There are many similarities as well as differences between the simulation development process and the software development process. The major difference is that, unlike in software engineering, the simulation model developer, nowadays, does not necessary need to write his/her own program code. Instead, predefined functions and block features inside the simulation software can be used. Apart from these major differences, all other features in both processes remain the same. Therefore, the study of the software development life cycle and the comparison it with the simulation project are indispensable in order to understand how the documentation of software development functions, and to identify the potential inputs for the documentation in simulation.

2.7.1 Software Development Process (SDP)/Software Life Cycle

Similar to the simulation project procedure, the study revealed that there are different approaches to the software development process, but two process models are currently in favour: the waterfall model and the spiral model (Abrams and Zelkowitz, 1995).

The waterfall model (Fig. 2.14) visualizes the software development process as a linear or sequential process upon a set of deliverable artefacts. In fact, it is one of the first models proposed for the software development process (SDP) (Pfleeger, 2001). The stages of SDP are generally sequential, but they are iterative between the subsequent stages. The feedback paths in subsequent stages represent knowledge gained in later steps. This knowledge influences the activities and the decisions made beforehand. Therefore, it may be necessary to adjust, or even abandon, because of feedback. Nevertheless, as in Fig. 2.14, there are easily recognized milestones between the steps in the process of waterfall approach. These milestones together with stages provide evidence not only for documentation but also for review stages in the whole process.

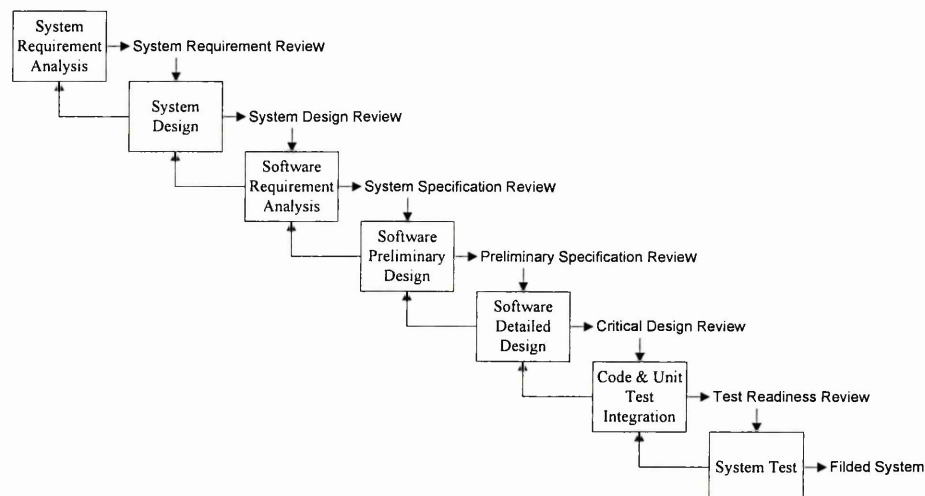


Fig. 2.14 : Waterfall Model in Software Life Cycle (Abrams and Zelkowitz, 1995)

Because of the deficiencies in the waterfall model, mainly in respect of iteration and risk, Abrams and Zelkowitz (1995) state that the spiral model has been gaining in favour. The spiral model emphasizes the repetition of basic activities at the progressive stages and accommodates changing requirements easily. Despite the advantages, the spiral model does not display much evidence about the details to be documented.

2.7.2 Milestones in Documentation in SDP

The documentation in SDP includes materials about the system specification, design, implementation, verification, installation, and so on. In many large systems, it may extend to dozen of volumes of text (French et al., 1997). Therefore, it is not feasible to discuss those details. Instead, it is considered relevant to understand how the layout of documentation in SDP is presented and what inputs are available for documentation in simulation. In this respect, the authors, such as Ayer and Patrinostro (1992), French et al. (1997), Basili and Abd-Ei-Hafiz (1996), Spear (1984); Barker, (1998) and Vliet (2000) present the following features that may be useful for documentation in simulation.

- The software development process or Software Development Methodology (SDM) is broken down into phases, sub-phases, and tasks for documentation purpose.
- Accomplishments that describe the design process and the results of each task is recorded and delivered into a documentation database as task deliverables. This process is referred as 'task-orientation' or 'task-oriented approach'.
- The format of the task deliverables plays an important role to maintain the clarity, consistency, uniformity, and understanding of documents.
- Texts in the narrative approach in task documents must be carefully prepared for easy understanding and for avoiding any ambiguity.
- The use of reusable program components makes the documentation much easier.
- Relationships between various task documents are to be established for data sharing.
- Icon and menu support for on-line task documentation is necessary for easy handling and updating task documents. For this purpose, CASE tools or other documentation tools that facilitate graphics as well as text capabilities are used with or without direct interfaces.
- The survey conducted by Forward and Lethbridge (2002), has declared that 54% out of 56 respondent cited that they use MS Word or other word processor for software documentation. However, 15% of respondents have said that they were least useful. In the same survey, Visio also was also identified as a useful tool for documentation.
- Accumulating and arranging task deliverables at the end of each phase of the development facilitate to prepare phase-end documents for a particular audience.

- Task documents from each phase are delivered into respective sub-databases for retrieving the material for phase-end documentation for different audiences.
- Documentation is extended for software development and planning and management information.
- Documentation contributes not only to the recording software development process but also to the production of operations guide, user guide, software development guidelines, etc. to facilitate for different audiences in software engineering.
- Software should be documented, as the system is being developed, but not retrospectively.

It is apparent that most of the above features can be seen or traced in the simulation model development process also. It may be particularly true for task documents that can be extracted by breaking the simulation model development process into practical micro-level of tasks. The task-orientation concept may fulfil many requirements of the documentation process in simulation. It may even be used as a communication media at review stages or as a progress report in a simulation project.

2.8 Documentation in Other Modelling Areas

Documentation is also applicable in other modelling areas such as mathematical modelling. One such documentation found is in ecological modelling, where mathematical models are gaining increasing importance (Benz et al., 2001; Benz and Knorrenschild, 1997; Benz et al., 1997; and Hoch et al., 1998). Mathematical models in ecology and environmental protection systems represent a large collection of scientific knowledge and experience. According to Benz et al. (2001), Benz and Knorrenschild (1997), Benz et al. (1997) and Hoch et al. (1998), this knowledge and experience is generally distributed across various documentation sources like journals, manuals, etc. However, such documents are available in different layouts and haphazardly linked due to non-availability of standard etiquette for documenting models. Hence, reuse of them is inefficient. ECOBAS system provides the solution for such a problem with a standardised structure and syntax for complete and consistent documentation of mathematical models as well as to make them accessible and comparable. This system facilitates the exchanging the information between the documentation model and executable model. It also translates documentation into text formats like HTML, TEXT, ASCII, and PDF. The complete approach emphasizes the importance of standardization for an efficient model development and model reuse.

The basic idea behind this approach is to break complex models into subcomponents (processes or objects) that may be used to build new models, to modify existing models and to facilitate recognition of possible reusable generic pattern. Then, the objects are linked

with classes, data, and ecological domains that represent the details and information required for documentation. In fact, this concept shows features of the OO paradigm. Objects can be linked, not only to present the model description, but also to represent model programming function for coupling with simulation system. Undoubtedly, exchange of information in this way makes the documentation more popular.

In contrast to the simulation model, the ecological model is concerned with the mathematical behaviour of the environmental factors. Therefore, the input from this approach to this study is limited. However, decomposition of the model into sub-components, and linking components for documentation and coupling components with graphical simulation models are worth being considered. At the same time, it offers evidence for the necessity of a common database for sharing or reuse of models.

The review of documentation in an ecological modelling system shows the significance in exchanging information between the implementation model and documentation model through standardized model components. Therefore, a similar approach may be appropriate for an integrated documentation system in simulation.

2.9 General Documentation Attributes

Documentation attributes, such as methods of presentation, styles of writing, grammar, etc. maintain the quality of any document (Forward and Lethbridge, 2002). However, the choice of the majority of these attributes is associated with the audience, who will benefit from the documentation. Although, these attributes are separated, they are interconnected to produce quality documents. On the other hand, the attributes, except the methods of presentation, are handled during the documentation stage whilst the method of presentation can be dealt with before hand. Therefore, the following discussion briefly outlines the options available for the document developer when they present the information in the development process. Methods of presentation for any document can be seen in two types: the narrative approach and the form approach (Ayer and Patrinostro, 1992; Russo, 1997, Barker, 1998). The decision regarding the method of presentation should be made before the documentation process gets underway. Regardless of the decision, the content remains the same in both approaches.

- **Narrative Approach:** This approach is the most commonly used format in documentation. It uses a free-form text narrative based on a standard outline prepared before hand. When this is applied, the following aspects must be considered during the document planning process.
 - Collect the material required for presentation and organize them to coincide with a standard outline prepared.

- Determine the user (audience/beneficiary) level (ISO Quality 9000 recommends writing at the lowest literacy level possible to facilitate the different levels of audiences) and present them.
- Use familiar terms and abbreviations or explain them thoroughly, wherever necessary in appropriate points.
- Break paragraphs into single ideas, use correct grammar, and keep it simple.

Despite the popularity, the document producer in the narrative approach does not have the luxury of pre-structured forms in manual or in-screen format to guide the contents of the development process. The challenge is further increased if the materials available are haphazardly organized.

- **Form Approach:** Unlike the narrative approach, this approach centres on the use of pre-structured forms for potential components and items in documentation. The forms can be formatted to present various types of details; text or graphical, by means of tables, flowcharts, algorithm etc. Like in the narrative approach, the functions of the audiences to which communication is directed should be considered in this approach. In comparison, form approach has the following advantages over the narrative approach to decide the suitability of forms for documentation in simulation.

- It produces uniform and consistent documents.
- Its pre-determined tables of contents can serve as a checklist for the documentation.
- It provides a measure of control for ensuring completeness of documentation.
- Even support staff can contribute to preparing the documents.
- Documentation can be prepared concurrently while the process is functioning.

Both the narrative and the form approach can be adopted for on-line documentation, with documentation tools such Word, Text Editors, Visio, or similar tools. In adopting the forms approach, specific consideration may be given to the development of menu-driven documentation system so that document presenter enables to call up the screen structures for document preparation. However, even for the narrative approach, the writer should be supported with necessity guidelines with possible menu structures.

Therefore, the adoption of forms approach, wherever possible, would be more appropriate to maintaining uniformity and consistency of records for the documentation framework being proposed in this study. As an additional advantage, such forms may act as a checklist for performing any simulation project.

2.10 Summary

The review presented in this chapter has focussed on the following evidence or has made the following choices:

- As discussed in Section 2.2.2, variation in project procedures shows that a revised simulation project procedure is needed for the documentation in simulation.
- According to the discussion in Section 2.3.4, integration interfaces of simulation software may be useful in developing an integrated documentation system.
- The review of reuse spectrum (Section 2.4.1) indicates that basic components reuse should be addressed for model representation or model documentation. Sub-model or completely model reuse may be enhanced through documentation.
- As per the discussion in Section 2.4.2.3, it appears that the separation concept may be applicable for creating software independent and reusable model elements.
- According to the comparison made in Section 2.5.2, flowcharts may be considered as the most appropriate technique for model documentation for the purpose of servicing different types of audiences.
- It seems that the task-oriented approach that was briefed in Section 2.7.2 can be adopted for documentation in simulation.
- Similar to the review presented in Section 2.8, regarding the documentation of ecological models, exchanging information between the implementation model and the descriptive model through a standard of model components may become the documentation in simulation popular.
- As discussed in Section 2.9, much effort needs to be made to propose a form approach for task documents to maintaining uniformity and consistency of records.

This review also raised the following key issues that need to be addressed in this research:

- What is meant by documentation in simulation? Documentation in simulation will be defined in Chapter 03, in establishing the documentation framework.
- Why is documentation in simulation necessary? Who benefits from documentation in simulation? These questions will be addressed, considering the individual needs of audiences that are benefited from documentation, and be presented in Chapter 03.
- What are the details to be documented in a simulation project? The details will be identified in Chapters 04 and 05 according the proposed approach for documentation.
- What are the details required for individual beneficiaries? This will be addressed in Chapter 04, after the details of simulation projects are identified and categorized.
- Why don't the modellers document the simulation project, or why do they do it idiosyncratically as they desire? Evidence shows that the answer is 'lack of structured methodology', but later this will be substantiated in Chapter 07.

3. RESEARCH METHODOLOGY AND ESTABLISHING DOCUMENTATION FRAMEWORK

3.1 Introduction

This chapter primarily aims to defend the methodology employed in the study and describe the foundations laid for establishing and evaluating the documentation framework. It begins with analysing the key issues raised during the literature review and on the general survey conducted for rationalizing the research topic. Hence, research questions for the study are set-up in keeping the focus of the study. The research approach employed and the strategies adopted to answer the research questions are outlined next, placing the work in the context of the research process discussed by various authors. The steps implicated in the research process are followed then for defending the rationale behind the selection of the research strategies. The documentation framework is also established in this chapter for setting out the groundwork required for the methodology. Based on the findings in the literature, characteristics incorporated in documentation process in simulation projects are identified as a first step. A new notion and approach are proposed for documentation, followed by the presentation of rules for illustrating how project details are created and managed. Finally, a revised simulation project life cycle is proposed for the purposes of documentation.

3.2 Research Questions

The literature review conducted on the existing simulation project procedures reveals that the documentation in simulation is an essential activity in simulation projects. The documentation in simulation delivers many benefits in the short-term and long-term (Section 2.6.2). However, it has been placed in different stages of the simulation project life cycle and interpreted in varying contexts and perspectives. These different viewpoints have raised the following research questions about the fundamental context in documentation:

Research Question 01 : What is documentation in simulation and why is it necessary?

Objective : To define documentation in simulation to fulfil different purposes in documentation.

As discussed in Section 2.6.2, many benefits from documentation have been recognized with respect to the audiences in the simulation project as well as model reuse. The review conducted on present practice in the documentation of simulation projects disclosed that the modellers pay a little attention to recording useful details. Whenever the documentation is created, modellers do it idiosyncratically with varying degrees of details. Such an approach may not fulfil the needs of individual audiences that are interested in the project. This situation has raised the second research question:

Research Question 02 : What are the details that need to be documented in simulation projects?

Objective : To identify the details to be documented in a simulation project to serve different needs of different audiences.

The views obtained from simulation practitioners and software developers revealed that non-availability of a structured approach to documentation is one of the reasons for poor attempt for documentation. The findings in literature regarding the strengths and weaknesses in existing documentation methodologies and the capabilities apprehended in simulation software for documentation (Sections 2.6.4 & 2.6.5) provide such evidence. Literature also suggests that treating the documentation process and model development process as two separate processes has made the documentation in simulation error-prone and less popular (Benz et al, 2001). This present situation raises the third question:

Research Question 03 : How should the documentation of simulation projects be accomplished?

Objectives : To develop a methodology for documentation of simulation projects and to make it an integral activity of the project.

The solution(s) to above research questions would focus the research process to achieve the aim of this research ‘developing a methodology to document simulation projects in manufacturing applications’. According to the aim of the research, it is clear that the research is focussed on manufacturing application. Nevertheless, the theory that needs to be created should be able to applicable to simulation in general.

3.3 Research Methodology

The research methodology is composed of a combination of research approach and research strategy/strategies, based on the nature of the research. The proposed research methodology is depicted in Fig. 3.1.

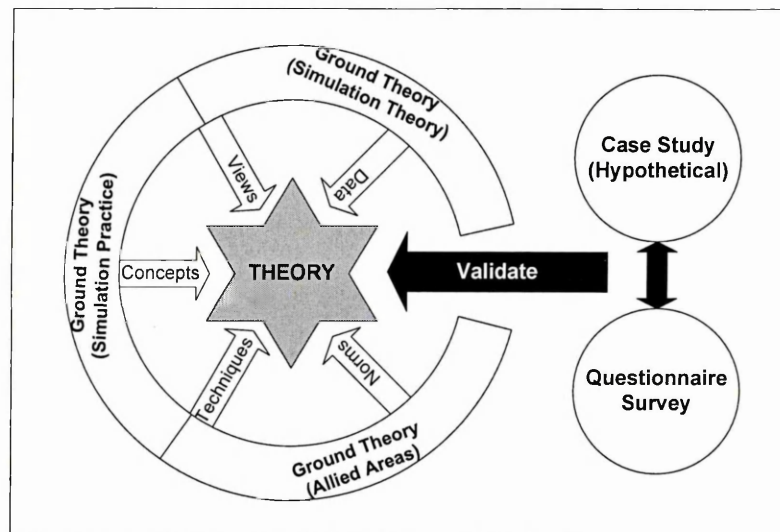


Fig. 3.1: Research Methodology

The rationale behind the selection of the research approach is determined by the analysis of research questions set up previously. The research approach was resolved by means of 'inductive approach'. The study of objectives in research questions directed the selection of the research strategy and it was determined 'grounded theory' strategy for collecting data to generate the theory. 'Questionnaire survey' supplemented with a 'case study' (hypothetical) is proposed for evaluating and validating the theory or techniques generated from the study.

The underlying principles of the selection process of research approach and research strategies are explained below together with the overview of the general research processes and their background.

3.3.1 Research Approach

Approaches to a research process can be seen in literature in two distinctive forms: deduction and induction (Saunders et al., 2003; Gill and Johnson, 1991; Gilbert, 1979). In deduction (also known as testing theory), the theory, or hypothesis is developed, and then is tested by an appropriate research strategy. This type is generally is considered as scientific research. Hence, it is the dominant approach in natural sciences, where the laws provide the basis of explanation and predict their occurrence. In contrast, in the induction approach (building theory), a theory is developed by analysing data collected and/or a technique is deduced as an application of the theory developed. Therefore, this is the dominant approach in social research. The characteristics in deductive and inductive approaches can be seen in quantitative research and qualitative research respectively as described by Locke (1998) and Taylor et al. (1984). The Fig. 3.2 summarizes the major differences between these approaches.


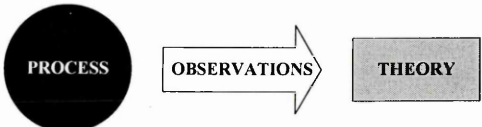
Deduction (Deductive Approach)	Induction (Inductive Approach)
	
♦ Understanding of scientific principles.	♦ Understanding of the human attached events.
♦ Explanation via casual relationships between variables.	♦ Explanation by understanding.
♦ Moving from theory to data.	♦ Moving from data to theory.
♦ Generation and use of quantitative data.	♦ Generation and use of qualitative data.
♦ A highly structured approach.	♦ A more flexible approach to permit changes.
♦ High concern about generalization.	♦ Less concern about generalization.
♦ Research Strategies: Experiments; Survey	♦ Research Strategies: Survey; Grounded Theory; Case Study; Ethnography; Action Research

Fig. 3.2 : Induction vs. Deduction in a Research Process

However, Saunders et al. (2003) and Gilbert (1979) emphasize that a clear adoption of one approach to a particular research is impossible in reality. They recommend that the combination of the two approaches to research is often advantageous. In support of this

argument, Kolb et al. (1979) describe that any individual may start the research with an experience of an event and then reflects upon that experience leading to the generation of explanations. Such a generation can then be utilized to form a theory that is generalised later for applying to similar events. Alternatively, the researcher can begin the study with the rules from other studies and subsequently applied and tested through their own strategy or strategies. The latter describes the characteristics incorporated in the deductive approach whilst the former illustrates the induction. However, in both cases, testing the rules in new situations offers new experiences. This new experience delivers observations and reflections that would result revisions to the rules tested and the generation of new explanations. Hence, the research process finishes with new rules. This brief description emphasizes that the selection of appropriate research approach is important at the beginning of any research.

It is clear that the research questions in Section 3.2, expect the answers for ‘what?’, ‘why?’ and ‘how?’ types of questions. According to Gill and Johnson (1991), and Yin (1994), these types of questions are the characteristics of the inductive approach. Therefore, the inductive approach with qualitative research strategies is the most suitable for employing in this research. However, it should not be forgotten that the research is also blended with some deduction, with respect to the first two research questions, which lay a sound foundation for answering the vital last question.

3.3.2 Research Strategy

The research strategy generally displays the overall plan for how the research questions are answered. Therefore, the research strategy employed should have clear objectives, which are derived from the research questions, a clear plan about data collection and well-thought constraints. According to Locke (1998), qualitative research, where this research is involved, engages in investigating observations in the environment or the process of the subject of analysis. This type of investigation is performed through selected qualitative research strategies such as survey, actions research, grounded theory, case studies as listed in Fig. 3.2. These strategies generate qualitative or descriptive data in the form of people written or spoken words and observations. Analyses of these observations form the basis for concepts or theories. With close examination of the objectives of the research questions set previously and the options available in qualitative research strategies, it appears that ‘grounded theory’ is more appropriate for collecting data and establishing theoretical framework in this study. Once the theoretical framework is established, a ‘questionnaire survey’ supported by a ‘case study’ with real-life or hypothetical context is suitable for evaluating and validating the theoretical framework.

3.3.2.1 Grounded Theory for Data Collection

According to Saunders et al. (2003), grounded theory is often thought as one of the best examples for the inductive approach. In this technique, data collection starts without formation of an initial theoretical framework. Theory is developed from analysis of data generated by a series of observations. Qualitative analysis on data gathered is conducted in a general sense with a less formalised or procedural way. However, a systematic and rigorous approach is still needed to arrive at explanation or theory. Such a data analysis leads to generation of predictions that are then tested for further observations which may confirm, or otherwise, the predictions. In this sense, grounded theory strategy inclines to a deductive nature but more biased to the inductive approach. This research involves well-established theories and practices in the discrete-event simulation processes of which facts and observations are well recorded. Hence, 'grounded theory' strategy is more justifiable for data collection, as it makes the provision for researcher to constantly compare the data collected and concepts used to aid the process of developing an emerging theory or a theoretical framework for documentation.

3.3.2.2 Action Research

According to Gill and Johnson (1991), action research generally is undertaken to solve a specific problem and at the same time to generalize from specific issues to contribute to theory. In this context, action research is considered as an inquiry or a research in the context of focussed effort to improve the quality of an organization or performance of a new approach. In support of this fact, Locke (1998) points out that it gives researchers new opportunities to explore and test new ideas and methods; to assess how effective the new approaches were; to share feedback with fellow researchers; and to make decisions about which new approaches to consider for implementation. Hence, it appears that the theoretical framework developed with grounded theory can be applied to different situations to assess how effective the new approach was and obtain the feedback for further improvement of the hypothesis or framework. However, such an approach increases the scope of the study. Therefore, an alternative strategy needs to be investigated.

3.3.2.3 Questionnaire Survey for Validation

In terms of continuum of research strategies, survey occupies an intermediate position between ethnography and experimental research (Gill and Johnson, 1991). Such a difference takes place due to the nature of the survey that depends on the intention and disposition of the researcher. Analytical surveys, which attempt to examine and explain the relationship between variables in the study context, have a deductive basis and they generally share the characteristics of scientific research. In contrast, descriptive surveys,

which are undertaken using attitude and opinion questionnaires and questionnaires of organizational practices, enable the researcher to identify and describe the variability in different phenomena. The use of descriptive surveys using open-ended questions to collect data and to explore a substantive area is often more ethnographic in orientation. However, according to Gill and Johnson (1991), much research may begin with an unstructured and exploratory investigation using explicitly ethnographic strategies. Such an approach shows the characteristics of the grounded theory. Therefore, the theory developed inductively with the grounded theory is to be tested later using a more structured questionnaire as part of the main study. In this respect, such a process may be considered as evaluation and validation of the hypothesis developed. It seems that action research is the best approach for validation of the theoretical documentation framework. Nevertheless, inability to use an action research within the scope of the study, structured questionnaire strategy is proposed for validation purpose. Structured questionnaire is supported with a case study to illustrate the new approach and/or explicated with structured interviews.

3.3.2.4 Case Study for Supplementing the Questionnaire

Case study research is the most common qualitative method used in information systems (Alavi and Carlson, 1992). Although there are numerous definitions, Yin (1994) defines the scope of a case study as an empirical inquiry that investigates a contemporary phenomenon within its real-life context, especially when the boundaries between phenomenon and context are not clearly evident. Case studies can be single or multiple-case designs, where a multiple design must follow a replication rather than sampling logic. When no other cases are available for replication, the researcher is limited to single-case designs. Yin (1994) pointed out that generalization of results, from either single or multiple designs, is made to theory and not to the populations. Multiple cases strengthen the results by replicating the pattern matching, thus increasing confidence in the robustness of the theory. Therefore, a single case study that illustrates the application of a new theory developed for a real-life situation seems to be more appropriate as supplemented strategy for validation process of the theoretical framework through a questionnaire survey.

3.3.3 Steps in the Research Process

According to the grounded theory strategy, the data collection begins with analysing the subject being studied. One of the sources suggested for data collection is from existing theory and literature. Simulation is considered as a well-established process that has had standard practices and procedures since its inception and it is continuing to develop in various directions. Much literature is available on those accepted procedures and their applications. From a documentation point of view, any documentation methodology should

agree with those standard procedures and practices and be flexible to accommodate new development. Therefore, as grounded theory, the most appropriate way to accomplish the research process is to study the existing literature for data collection. A qualitative analysis of data is then carried out to derive relationships within the initial data in the context of this research. Initial data are collected by studying literature for the following steps with specified purposes:

- A review of literature in general to understand the state-of-the-art in simulation.
- Analyse of typical simulation project procedures:
 - to understand the current context and practices in simulation project,
 - to propose a standard project procedure for documentation,
 - to investigate existing strengths and weakness in documentation,
 - to identify details to be documented in simulation projects, and
 - to propose a standard structure for records.
- Review of literature on manufacturing application to add real world characteristics into the details that to be documented.
- Investigate the exiting documentation methods in simulation and other allied areas to identify the strengths and weaknesses; and to propose a methodology for documentation of simulation projects.
- Examine the literature on reusability to create software-independent and reusable model elements.
- Study the documentation standards to identify general attributes in documentation.

The above steps display the detailed view of data collection through grounded theory strategy. The major steps immersed in the strategies adopted for the complete research process is briefly outlined in Fig. 3.3. As depicted in the figure, the rectangular boxes illustrate the major steps in the research process. Shaded boxes in the middle of the figure show the outputs from each step, which subsequently will produce the Model Representation and Documentation (MRD) system and tool, which are represented by a circle. Rectangular boxes on the left denote the sequential steps in the research process and those on the right represent the generalization and reinforcement of the results in each step.

The grounded theory strategy makes provision for researcher to constantly compare the results of the research process and refine the theory developed. Therefore, results from the validation process, which are supported with expert views and application of the methodology for a real-world application, are accommodated through the feedback path for the results in immediate steps, as depicted in Fig. 3.3.

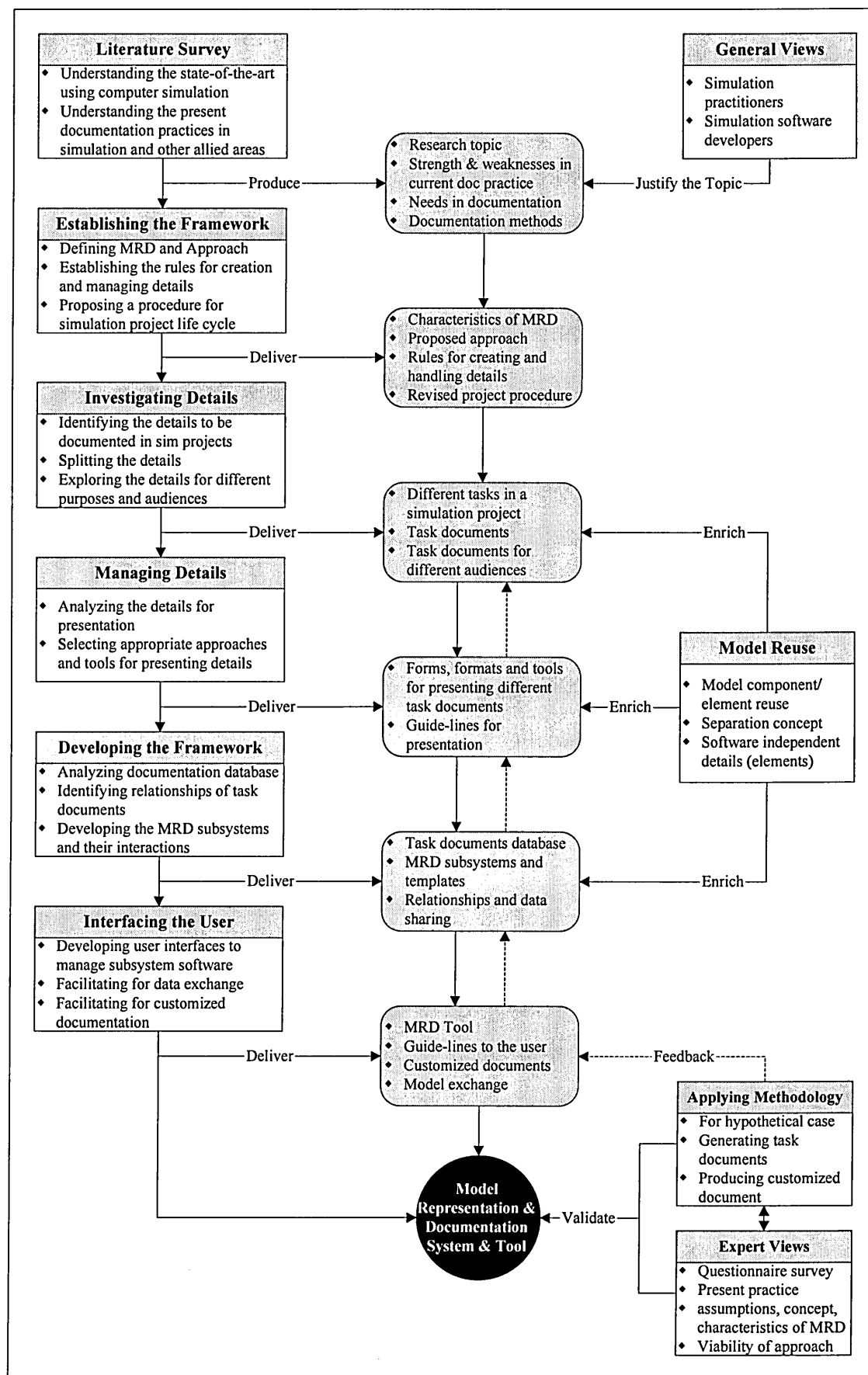


Fig. 3.3: Steps in the Research Process

3.4 Establishing Documentation Framework

The development of a methodology for documentation in simulation is based on the sound establishment of a theoretical framework for documentation. The characteristics, strengths, and weaknesses in present context and practice in documentation in simulation, and the needs of documentation for different audiences are the basis for establishing the framework. Therefore, the characteristics expected in the framework to be established are investigated, followed by a new notion and approach for documentation of simulation projects.

3.4.1 Characteristics of Documentation Framework

Within the current context and practice in documentation in simulation, the following characteristics that are found in literature can be considered as useful in establishing the framework and achieving the aim and the objectives set out in this research.

3.4.1.1 Benefiting Audiences

The literature shows many benefits from documentation in simulation. They can be seen in two broad perspectives: the short-term benefits (for the project) and the long-term benefits (for model reuse) as discussed in Section 2.6.2. Analysis of those benefits, with respect to beneficiaries, reveals that audiences that are benefited from documentation can broadly be categorized into three levels: model users (mode owner), model re-users, and other interested parties. This categorization is a further expansion to the classification made by Oscarsson and Morris (2002). A clear of level distinction between model re-users and other interested parties is not clearly visible as later may become the former, in time, but they are considered separately for analysing the needs of different audiences and for establishing a more flexible framework. With this view, the documentation framework to be established should be able to fulfil the needs of individual audiences that are discussed briefly below.

➤ Model Users (Model Owner)

Model users, in the context of this research, are not the model's executors, who conduct the experiments. They are the personnel who are involved with, contribute towards, or are interested in the current project. They typically are the client, team leader/project co-ordinator, modeller, system supporter, model supporter, data provider, model executor, and interested in-house members. This classification is a slight expansion to the team proposed by Robinson (1994). The names of the categories show different responsibilities and the needs of individuals in the project. Therefore, the responsibilities of these individuals together with their needs are briefly explored.

- **The Client:** The client is the project sponsor and could be a single person, like a manager, a director, a consultant, or a group of people, like a board of directors, who make major decisions regarding the direction of the project. As they are the

recipients of the results of the project, project objectives, and how they are achieved within the constraints available, are the important details for them. Therefore, their understanding on overall progress of the project is important.

- **The Team Leader/ Project Leader/ Project Co-ordinator:** The team leader, the project leader, or the project co-ordinator is the one who drives the project to achieve the target set within the time-scale. The team leader typically does not necessarily have to be involved with the modelling process or to be aware of the project details in depth. Instead, he/she needs to monitor the progress of each task in each phase of the project, keeping the enthusiasm of the team high and ensuring that a consensus is reached. It is his/her responsibility to inform the client and the other interested parties about the status of the project.
- **The Modeller:** The modeller builds and tests the simulation model contributing for most of the tasks, from the beginning of the project. Therefore, in depth understanding of every details of the system being modelled is essential for the modeller. With this view, almost all types of details in a simulation project could be considered as significant for the modeller to perform iterative natures of the model building process. This may not seem to be true if more than one modeller share model-building process in a complex project. However, as the tasks in simulation project are generally in iterative nature, the understanding of details as much as possible helps the modellers to achieve the ultimate targets of the project. The modellers are expected to have in-depth understanding of most of the tasks in a project, but they cannot be experts on the system being modelled or may not have the necessary skills on the simulator being used. In such a situation, an additional support may be necessary for understanding the system and modelling.
- **The System Supporter:** In house members, who are experts on the system or problem being studied, are sometimes invited to contribute to the system investigation, model representation tasks, etc. In such instances, the project objectives, model scope and level of details etc, depending on the purpose that they are invited for, may be relevant to their effective contribution.
- **The Modelling Supporter:** In some instances, modelling support is sought from an expert from the software vendor for model coding and testing. As their role is only for modelling, model description is sufficient for them to complete the task.
- **The Data Provider:** Data collection for a simulation project needs a great attention as the collection process takes a considerable time due to many constraints. Therefore, a separate data provider, who is generally an expert in the system being modelled and either has direct access to data, or knows where they

can be found, is required for the smooth running of the project. In order to deal with the data collection process successfully, the data provider needs to be aware of the required data, data format, time-scale, and perhaps the use of data.

- **The Model Executor:** The model executor who generally is referred to as model user in literature (but, who, in the context of this research, is the model owner, including the model executor), performs experiments, analyses, and presents the results. For these purposes, it is important for her/him to understand what the experimental factors are, how they are changed in different scenarios, how experiments are conducted, what results are to be gathered, and how results are presented for decision-making process. However, he/she does not need to know details of the model, though awareness of some detail may be useful for conducting experiments and analysing the outcomes.
- **Interest In-house Members:** Sometimes, members of the staff, like managers, engineers, etc. in the organization show their natural vested interest towards the entire project or part of the project. They may interact with the model development process, though they are not directly involved in the team or the project itself. Therefore, they may need reviews of progress, demonstration of the model and conclusions and recommendations of the project, depending on their interest.

➤ **Model Re-users**

The reuse of previous concepts, knowledge, and models in simulation would result in a great deal of cost and time benefits (Kovacs et al., 1999 and Pratt et al., 1994). However, there are some arguments against this. Regardless of the views, reuse of an existing model can naturally be considered as an initiation from 'client/top-level managers' who have clear views of the organizational goals and constraints; and wish to broaden their horizon. Once the top management accepts the model reuse, 'analysers/modellers' explore the model to find out any alteration to the model structure, if required, to suit their particular application. Thereafter, the project associated with model reuse may be considered as a new simulation project, but with a little modification to the existing model and records to suit the new application, and the beneficiaries from documentation would be similar to that in model users. Therefore, the audiences that are benefited from documentation in model re-users category can mainly be considered in two sub levels: 'clients/top-level managers' & 'analysers/modellers'. Their responsibilities are self-explanatory from the model reuse point of view. In respect of their needs, the proposed documentation should provide details for 'clients/top-level managers' to explore the overview and benefits of the project; and for 'analysers/modellers' to review the model critically for any alteration.

➤ Other Interested Parties

In addition to model re-users in the second level, there may be some other interested parties, like students, researchers, who wish to use an existing model for the purpose of improving their knowledge and skills in simulation projects in general and specifically in model development. They may require different set of details depending upon their interest, but it seems that they need more technical details rather than management details. As stated previously, a clear level of distinction between this category and the model re-users, specifically analysers/modellers, is not apparent. However, they are still considered separately in order to establish a flexible documentation framework.

3.4.1.2 Progressive Documentation

The documentation of any project offers significant contribution to its success (Forward and Lethbridge, 2002). A simulation project is not an exception. It acts as a communication tool that enables the simulation project team and others who directly and indirectly involved, being aware of the project details and its progress, and avoiding early misunderstanding (Musselman, 1994). Therefore, it seems that the sooner the documentation is dealt with, the better the results would be. Nevertheless, it seems that the general trend towards documenting details is a retrospective approach - after the project completion -, except for a few details such as objectives, model description, whenever it is attempted. Such attempts may theoretically enhance the model reusability and knowledge passing, but, in nature, people tend to forget the important details or ignore them once the goal is achieved. Therefore, it seems that the documentation as the project progresses - progressive documentation - provides more benefits for the project.

3.4.1.3 Continuous Recording

It is an accepted fact that any document should encompass with important and minimal records. The significance of the record depends on the purpose of the documents and for whom it was prepared. However, a brief description presented about the needs of different audiences in Section 3.4.1.6 elaborates that diverse ranges of details are demanded for documentation and they spread throughout the simulation project. It is also evident that there are many overlapping needs for different audiences. On the other hand, no simulation projects may accommodate all types of audiences discussed previously. This situation may lead to difficulty in identifying the significant and minimal records during the documentation. Therefore, a random approach to documentation would be problematic or it may reverse back to retrospective documentation. Hence, it seems that the best approach is to record details continuously as the project progresses and then to search for significant and minimal requirements for different purposes and audiences. Such an attempt gives an indication about task-documentation.

3.4.1.4 Model Representation as a Core of Documentation

Many authors treat model representation as the only component in documentation or model documentation. A very few like Robinson (1994), Bank (2000), and Law and Kelton (1991) consider it as a key component in documentation. Without exception, it should be the vital component in the documentation framework to be established. However, it is obvious that simulation model development process engages, with not only model representation, but also other important tasks that deliver much more detail and knowledge embedded with the model and the process of model construction. Such details are vital for understanding the model and their outcomes, not only for participants in the current project, but also for future users of the model. Therefore, it is important to define the documentation in simulation to avoid any confusion before a documentation methodology is proposed.

3.4.1.5 Software Independent Documentation

Software is generally involved from the beginning of model development phase. However, the project team may tend to think about the software and its facilities even before the model development. Hence, the project development process and the records may be considered in terms of the software that is intended to use. Particularly in model representation (or model documentation), the typical practice is to produce flowcharts or block diagrams, which are biased towards the simulation software being used, to represent some aspect of logics (Schormann and Perera, 1999). The document users (in the case of model reuse or in other potential future users) may not be able to understand such records unless they are familiar with the particular software used. If the records are independent from the software being used and if neutral terms and techniques are used to present records, wherever possible, such generic details may enhance the understanding and the effectiveness of the documents.

3.4.1.6 Integrated Documentation

It seems that a proper documentation in simulation is generally considered as a time-consuming process. Therefore, it could be another burden or project for the project team to deal with, unless the documentation and the model development provide direct benefits to each other. In supporting to this same argument, Benz et al. (2001) states that treating the model development and the documentation as separate processes makes the documentation unpopular and error prone. Therefore, the documentation should be performed in such a way that the recorded details can also be employed for the implementation model, wherever and whenever possible. Hence, the documentation in simulation seems to be an un-isolated or an integrated process with the model development process.

3.4.1.7 Structured Approach

It is perceptible that the effective development of a documentation framework is contingent to the establishment of a common layout and structure with systematically linked details of the simulation project. But, an extensive literature review on existing documentation methodologies (Section 2.6.4) and capabilities in simulation software for documentation (Section 2.6.5) disclose that non-availability of structured methodology for both internal and external representation of simulation projects, either in the manufacturing discipline or any other sector, is one of the major reasons for poor documentation. Therefore, any documentation framework should be based on a structured approach that facilitates for progressive, continuous, and integrated documentation.

Having investigated the characteristics expected of a documentation framework, the answer to research questions one can be seek to fulfil the needs identified in respect of different audiences. The answer would primarily create the necessary background to address the subsequent research questions, and lay a sound foundation for developing a methodology for the documentation of simulation projects.

3.4.2 Research Question 01: What is Documentation in Simulation? Why is it necessary?

The varying context and perspectives in documentation in simulation and the significance of model representation activity in documentation have led to a fresh notion. Hence, documentation in simulation is named as ‘Model Representation and Documentation (MRD)’ to avoid misconception and to broaden the meaning. The necessity of documentation arises due to different needs in audiences that benefit from documentation. Accordingly, MRD can be defined to answer for both parts of the first research question.

3.4.2.1 Definition for Model Representation and Documentation

Having taken the needs of different audiences in the project as well as potential future usage into account, MRD is defined as recording the knowledge, information, and results embedded with each step in a simulation project in order to fulfil the following purposes:

- **Communication:** to communicate the state of the simulation project
- **Dissemination:** to disseminate the accumulated knowledge and experience
- **Motivation:** to keep the enthusiasm of the team high
- **Quality:** to ensure the quality and creditability of the model and the project
- **Reporting:** to report the results and findings of the project
- **Maintenance:** to guarantee the project progress or error traceable
- **Contingency:** to protect against time and knowledge losses (personnel turnover)
- **Enhancement:** to enhance the effective model reuse

3.4.2.2 Task-Orientation

The purposes of MRD can only be achieved through continuous recording of details of individual tasks as the project progresses (progressive documentation) and then accumulating them to fulfil the needs of different audiences. This is where the ‘task-orientation’, which is applied to documentation of the software development process, can be considered for recording details of the entire simulation project life cycle.

Task orientation can broadly be defined as an approach to MRD that presents details of the project in chronological order, based on the sequential tasks of a simulation project. However, the model development process is generally iterative by nature and it can never be a sequential process. Nevertheless, the chronological order of the activity steps is considered initially to identify the details to be recorded and the iteration process is left to be addressed later in the MRD tool. Within the task-orientation context, MRD in simulation can be viewed in the following two levels: ‘task documentation’ and ‘phase-end documentation’, according to the level of details comprised in a document.

➤ Task Documentation

Task documentation is defined as the recording of the accomplishment that describes the process and results of each task performed sequentially or iteratively during each phase of a simulation project. The output from task documentation is a group ‘task details’ and is referred to as ‘task documents’ or ‘task deliverables’. In this context, it appears that the continuous progress reports fulfil the purpose of task documentation. Although the progress reports give relevant history of the work been done in chronological order, they are not properly planned for a structured documentation system. Therefore, properly planned and designed task documents are important to establish a complete, systematically linked, tool-independent and mutually comparable documentation system. However, such a systematic recording could be arranged to fulfil the requirement of a progress report though it may not be true vice-versa. The task documents created during the simulation project are stored in a documentation repository for retrieving them later for next level – phase-end documentation.

➤ Phase-End Documentation

Phase-end documentation is defined as the vehicle of communication that accumulates and arranges task documents for a particular purpose or audience. In other words, it synthesizes and recognises the task documents for presentation in appropriate forms to individual groups concerned with the project. The documents are prepared by retrieving appropriate task documents from the documentation repository and then re-arranging according to the needs of different audiences.

➤ Norms for Creating Task Documents

Phase-end documentation raises the following issues regarding how the task documents are created:

- What are the task documents that are to be incorporated in a phase-end document for a particular audience?
- What are the task details that are to be included in a task document?
- Can a task document serve to different phase-end documents without alterations?
- What level of task details is to be incorporated in a task document to serve different audiences?
- What are the repetitive details that need to be shared with other task documents?
- How are task documents to be presented for easy understanding by audiences?

The above issues primarily are centred around creating individual task documents. Those issues can be addressed by breaking details of the simulation project life cycle into task documents in accomplishing the following norms and arranging them to produce different phase-end documents.

- **Modular:** Task documents should be flexible enough to serve for any phase-end documents without any alterations (adding or removing details) to task details.
- **Recurrence:** Any specific detail may be repeated to maintain the modularity, but keeping them at a minimum level or referencing with other task documents without affecting modularity.
- **Minimal and Concise:** Task details should be kept to a minimum level, but taking the different levels of audiences into account.
- **Generic:** Task details should be kept common terms and be independent from the simulation tool, wherever possible, to serve different level of audiences.
- **Exchangeable:** Task documents should facilitate exchanging details among the other task documents, and with the executable model, to share repetitive details.

➤ Norms for Managing Task Documents

Once the task documents are created according to the norms established, they can then be stored in a documentation repository for retrieving them later for phase-end documentation. This identifies up the following issues regarding how they are presented and how they are handled conveniently:

- In what form and format is the task documents presented?
- How and where are the task documents stored?
- How easy can the task documents be accessed and altered?
- What level of skills and training is necessary for the user to handle them?

The choices to be made for the approach, the media, and the tool to manage task documents provide the answers to those issues. Therefore, the choice can be made, accomplishing the following norms:

- **Uniform and Consistent:** Task documents should be presented using a form approach, wherever possible, to provide uniform and consistent information for any simulation project.
- **Simple:** In text presentation, task documents should be presented in a simple way to serve different levels of audiences.
- **Media:** The medium for handling task documents should be electronic for convenience of storing, retrieval, tracing, and altering the details.
- **Independence:** Tools should not be limited to any simulation software or documentation facilities accomplished with the simulation software.
- **Portable:** Tools to manage task documents should run in a common platform, like PC based - MS Windows.
- **Competence:** Tools should be more user-friendly and easy to learn without much training.
- **Scalable:** Tools should not limit the size of the task documents, i.e. allow scalable generation of task details, and task documents.
- **Exchangeable:** Tools should facilitate exchanging full or part of in task documents among the other task documents and with the executable model.

3.4.3 Revised Project Life Cycle for MRD in Simulation

In establishing a documentation (MRD) framework, it was emphasized that the task documents play a crucial role in the proposed approach. Creation of these task documents and development of structured methodology for MRD can only be achieved through the comprehensive study of the standard procedure of a simulation project.

However, the review conducted in Section 2.2 reveals that there are inconsistencies in sequential steps in the simulation project procedures. For example, Balci (1998) include the system investigation task in the project specification phase, before the model design and development phase starts, whereas, Tye (1999) considers it under the model design and development phase. But, both instances encounter the same task of analysing the system for better understanding of the problem, enabling the modellers to identify the elements and their details to be incorporated within the model. This situation raises the question 'which procedure should be adopted for documentation?' Investigating an appropriate approach for documentation will inevitably increase the scope and time frame of the research study. This situation shows the necessity of revising the simulation project procedure for MRD purpose.

As discussed in Section 2.2.2, the breakdown and terminology of the project life cycle that are found in literature differ from one to another. Nevertheless, there has been at least partial agreement that the phased-development of the simulation project should revolve around the following four broader stages, or phases, which are similar to project procedures proposed by Robinson (1994), Tye (1999) and Pidd (1989 and 1992).

- Identify project specifications
- Build and test the model
- Conduct experiment, and
- Implement the project

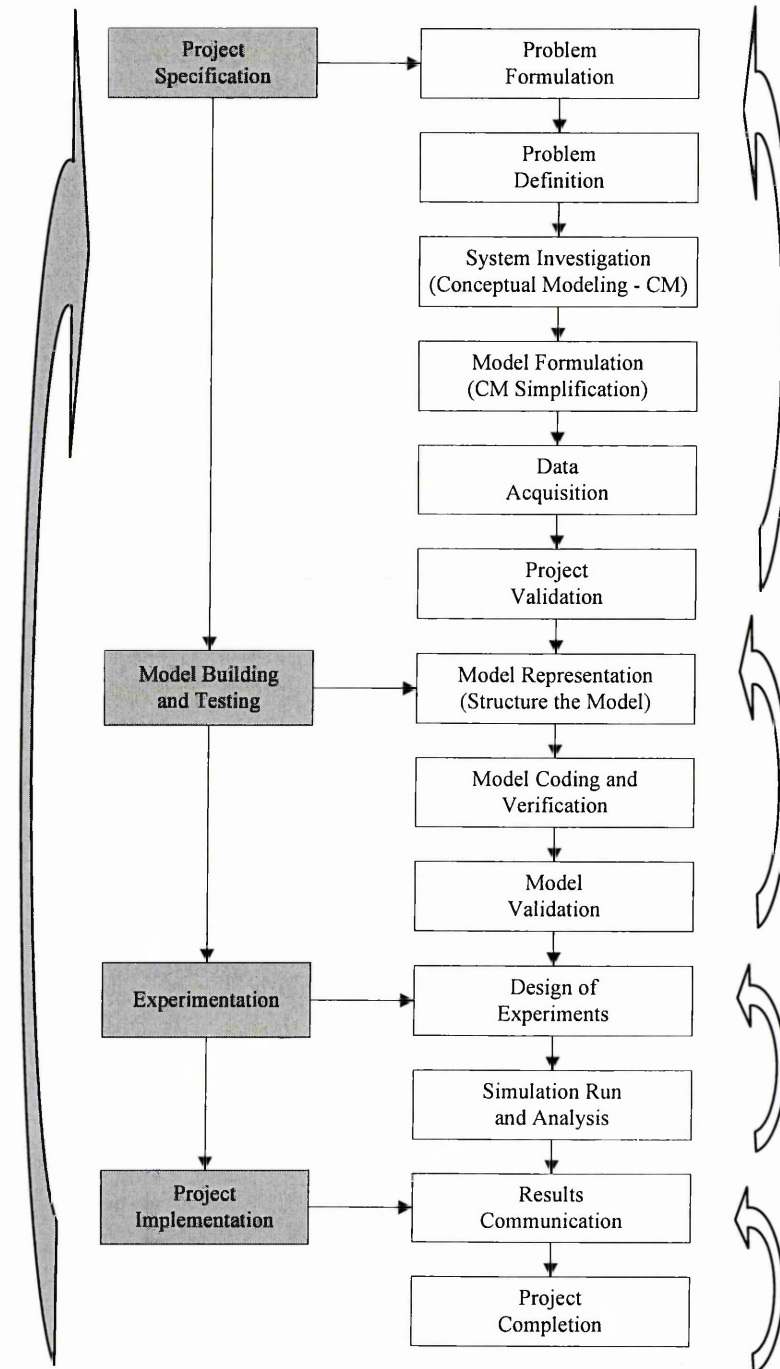


Fig. 3.4 : Revised Steps for Simulation Projects

As shown in Fig. 3.4, each of the broader phases can be considered as having sub-phases, in complying with detailed approaches proposed by Banks (2000), Law & Kelton (1991), Shannon (1998), Maria (1997), and Balci (1998). According to Law and Kelton's (1991) and Balci's (1998) approaches, the validation takes the prominent place in several different stages of the project life cycle. Therefore, the validation process is proposed at the end of the project specification as well as the model building and testing phases in the revised project life cycle. In the proposed procedure, the documentation stage cannot be visible as a prominent step incorporated in sub-phases. It is a parallel process rather than a single activity like in typical simulation project procedures proposed by Banks (2000), Law & Kelton (1991), Shannon (1998), and Maria (1997). With the proposed revised procedure, which shows much in common with the structure presented by Robinson (1994), the simulation process is broken down into 13 sub-phases in order to facilitate the task documentation. More about the sub-phases and their details will be discussed in Chapter 04.

As discussed in Section 3.4.2.2, the proposed approach to MRD is task-orientation. Therefore, a breakdown to a standard or revised project procedure is necessary in order to identify the tasks and then task documents together with task details. Such a breakdown to major phases, as depicted in Fig. 3.4, has been adopted with the view of identifying the sequential details that are to be documented. Subsequent breakdown of these sub-phases, which are discussed in Chapter 04, will lay a foundation for investigating the answer to the second research question – ‘What are the details to be documented in a simulation project?’ – based on the new notion and norms defined for documentation in simulation.

3.5 Summary

This chapter has summarized the research questions and the research methodology employed for answering those questions, followed by the research steps, which immersed in the research strategy adopted. Based on the literature review and views obtained from simulation community, it has also presented the characteristics that are to be incorporated in the documentation framework that needs to be developed. New notion for documentation, which is named as Model Representation and Documentation (MRD) in simulation, has been defined, based on individual needs of audiences that would benefit from documentation. This new definition provides the answer to the first research question. Finally, the chapter has concluded by setting up a novel approach – task-orientation - and norms, not only for the development of documentation methodology, but also for subsequent chapters. Consequently, Chapter 04 presents the details that are to be documented in simulation projects, based on the revised projected procedure for simulation project that has been proposed in this chapter.

4. TASK DOCUMENTS

4.1 Introduction

The objective of the second research question is to identify the details that are to be documented in a simulation project. This was raised due to different interpretation presented by different authors in respect of documentation and diverse need for different audiences. The details generally spread throughout the entire simulation project. As suggested in Section 3.4.2.2, the proposed approach to the documentation is the task-oriented approach, by which the details accomplished on each task of simulation projects are identified and recorded progressively. In this attempt, the grounded theory strategy was applied to relevant literature on typical simulation project procedures for the purpose of collecting data and comparing them with practical aspects as discussed in the research process (Section 3.3.3).

Therefore, this chapter primarily aims to present a comprehensive list of details that are to be documented in a typical simulation project. However, the investigation process of details is briefly discussed, taking each sub-phase and each primary task of a simulation project into account. Although the details are based on the literature, any emphasis is not given on specific literature due to their commonality. The chapter concludes by providing a compressive list of task documents. Nevertheless, the detail description of task documents and how they are managed for presentation will be discussed in chapter 05.

4.2 Primary Tasks in a Simulation Project

The proposed revised project life cycle for Model Representation and Documentation (MRD) in Fig. 3.4 exhibits two levels of activities in a simulation project. The phase level (the first level) represents the project activities in broader perspectives. The sub-phase level (the second level) signifies a detailed level of each phase. But, a further level (micro-level) of abstraction is necessary for fulfilling the task-oriented approach and complying with norms established for creating task documents as discussed in Section 3.4.2.2. For example, the problem definition sub-phase comprises of identification of project objectives, experiments to be conducted as well as reports to be generated. Therefore, each sub-phase of the project is examined in order to determine micro-level of activities. This micro-level of abstraction is referred to as 'primary tasks' of the simulation project. The analysis of the project life cycle produces 25 primary tasks, as illustrated in Fig. 4.1. Despite the fact that the activities in each level are shown in linear fashion, they are generally in iterative nature. This iterative process is captured in Chapter 06, once the task documents of primary tasks are comprehensively analysed.

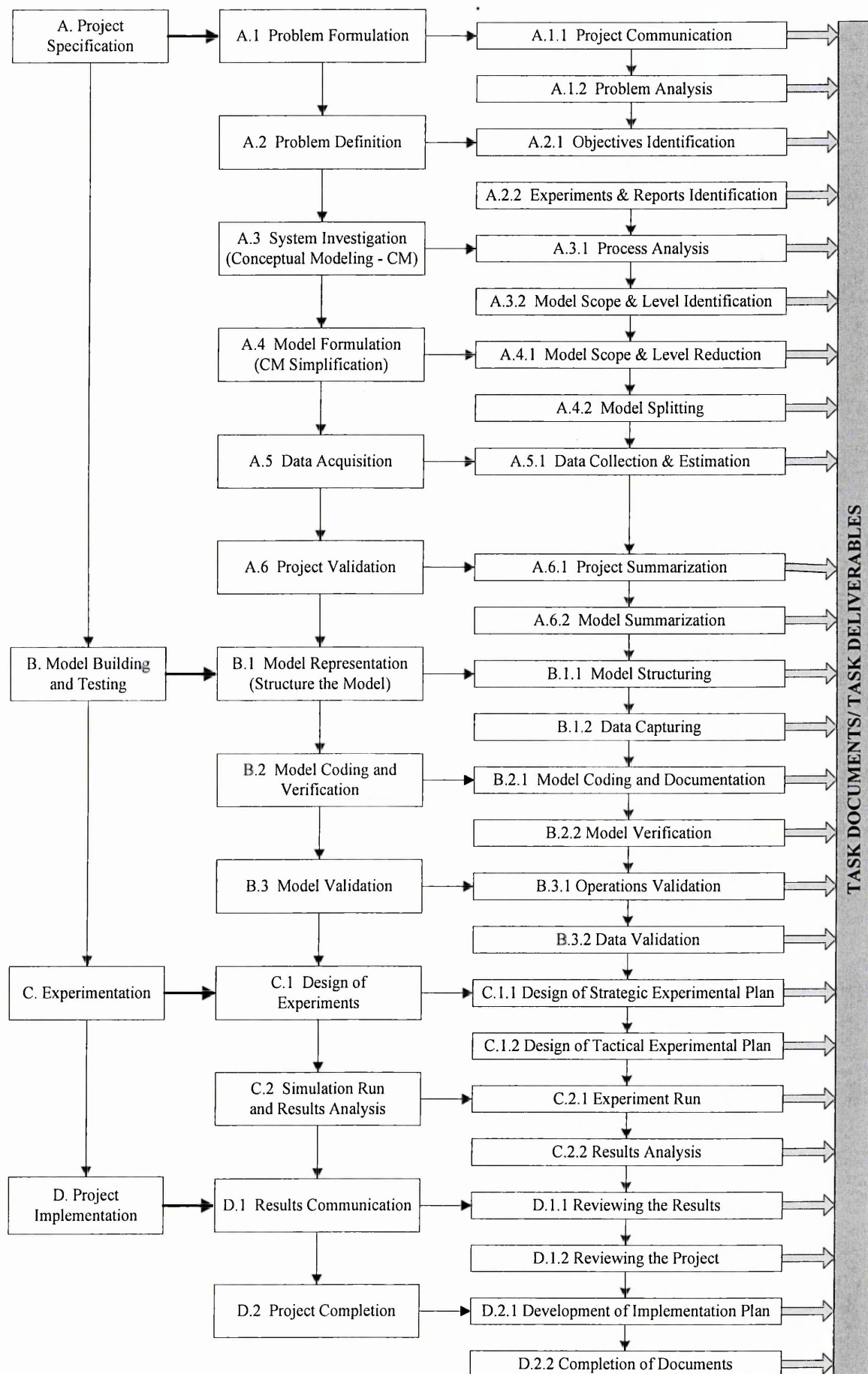


Fig. 4.1: Primary Tasks in a Simulation Project

4.3 Task Documents/Deliverables in Primary Tasks

Each primary task of typical simulation project is then examined to investigate the details to be recorded, in complying with the norms established for creating the task documents. The records that accomplish the process performed and the results obtained in each primary task are known as task documents or task deliverables. A primary task may deliver one or more task deliverables to the documentation database, depending on the nature of details. Such an approach facilitates more flexibility to retrieve appropriate details for different audiences (i.e. complying with modularity of task documents). The details are recorded and stored step-by-step, sooner the primary task is completed, and then recalled for phase-end documentation. Hence, this approach collectively establishes the requirements of MRD for the entire project life cycle. With an intensive analysis of the simulation project life-cycle found in literature, 40 task documents have been identified. Therefore, a brief overview of the analysis process of sub phases and the primary tasks, together with the list of task documents, are now presented. The concept adapted to separate reusable model elements for model representation is also elaborated in appropriate discussions or situations to support the documentation framework. Such reusable model elements have a certain influence on other task documents.

4.4 Task Documents in Phase 01: Project Specifications (A)

Different authors present different views and breakdowns of activities for this phase. Regardless of different views, the end results of this first phase should be the finalized project specifications that comprised of clear aim and objectives, experimental factors, reports to be generated, model scope and levels, assumptions and project plan so that the model building process could be performed with minimal interaction with this phase. It is also a fact that this is the phase where the experience of the project team plays a key role. They apply their knowledge and experience into practice without much attention on recoding. Subsequently, it may result untraceable project and lack of dissemination of knowledge and experience. Therefore, this phase takes very important place in documentation unlike other phases that deliver recorded details in certain extent.

To achieve the above requirements, project specification phase begins with understanding the problem through problem formulation. Then the problem is defined to identify the objectives and the experiments to be conducted. Once the problem is defined, an investigation of the system is carried out in order to develop the conceptual model, followed by the model formulation sub-phase to simplify the conceptual model. Data acquisition stage starts next to ensure that data is readily available for the model development. Finally, the project is validated to get the acceptance from all parties, involved in the project.

4.4.1 Problem Formulation (A.1)

A need for a simulation project brings by a client. The client's problem description is usually considered as a set of symptoms. It may have been described in terms of the client's background and the experience on the problem as well as the simulation aspects. However, Tye (1999) stresses the importance of analysing the system with someone who understands the problem and system well. This is because the operation of the real environment is often different from that, which is understood by the client or the management. Therefore, a better communication with the client is required to aware the need statement. Then, an analysis of the problem and/or system with relevant key personals involved with the study domain is essential in order to study the feasibility on the need statement. Gathered information from such analysis enables the project team to lay a strong foundation for the problem definition.

4.4.1.1 Project Communication (A.1.1)

Project communication is the task for the client to initiate the communication process with simulation expert or analyst about the request. The request might be either for a new systems development, for an enhancement to an existing system or a solution to a prevailing problem. It is naturally a formal document, comprised of self-description about the client or organization, a brief description about the problem and the benefits seek from the project. It may also be followed by verbal explanation. The client's view about the problem may not always analytical, depending on the client's in-house expertise. Once for a simulation project is sensed and an agreement is reached, a team to proceed with the simulation project needs to be decided. The project team may also comprise of client's in-house members too. In this respect, the project communication task delivers the following three task documents into the MRD system.

A.1.1.1 Overview of the Client: with the aim of presenting overall picture about the client. The task document may comprise of:

- Client details,
- Business goals in respect of the system under study,
- In-house expertise in respect of the system or problem, and
- In-house expertise in respect of the simulation concepts.

A.1.1.2 Problem and Need Statement: with the aim of recording the problem proposed and client's expectation, enabling the document user to compare with the problem diagnosed and additional benefits. The document may include:

- Brief description on the problem or system,
- Any solution techniques proposed,
- Any constraints, and
- Expected benefits.

A.1.1.3 Project Team: with the aim of highlighting contribution expected from members of the team and the reasons for the selection of individuals. This includes:

- Team members,
- Description of the members,
- Individual responsibilities, and
- Reasons for the selection of individuals.

4.4.1.2 Problem Analysis (A.1.2)

Problem analysis task allows the project team to study the problem in-depth, enabling them to lay a strong foundation to define the problem clearly. Generally, the client presents a series of symptoms, as being problems. Their view about the problem may not always correct, but they may hold a clear mind what they expect as outcomes. This emphasizes a need for reviewing the problem with relevant in-house members, if exists. Such analysis allows the project team to understand the nature of problem and/or system and the factors pertaining to the system and its environment that are likely to affect the solution of the problems; and to identify the potential outcomes. Subsequently, unnecessary bottlenecks are avoided at the early stage of the project. In this view, project analysis task delivers the following two task documents into the MRD database.

A.1.2.1 Form of Review: with the aim of illustrating the major issues reviewed in the problem formulation stage and how they were addressed. The record consists of:

- Methods of review,
- Personals participated,
- Issues reviewed, and
- Review summary.

A.1.2.2 Review Outcomes: with the aim of highlighting mostly affected outcomes for the issues reviewed and any deviation of cause and effects or system brought forward initially. This may include:

- Any alteration to the problem or system,
- Any alternative solutions to the problem,
- Any constraint,
- Any assumptions made in solution techniques, and
- Achievable project benefits.

4.4.2 Problem Definition (A.2)

It is a fact that for anything to be successful, it has to be directed towards a clearly defined goal. A simulation project also should have clearly defined objectives, which are communicated and agreed by the client and the simulation team. In this view, the

objectives, related to the problem should be defined in the first place in order to establish that which should be achieved and what knowledge should be gained, once the project is completed. Although the problem objectives make the client alive with expectation, the time frame of the overall project and constant progress of the project are some other issues on which they are interested. Therefore, objectives should be identified not only related the problem or system under study, but also in relation to the project time frame. The information required to achieve the problem objectives is provided through experiments. The success of these experiments is evaluated by measures of performance that are recorded on experimental reports. Therefore, deciding the experiments to be conducted and the reports to be generated in this stage clearly illustrates the direction of the project.

4.4.2.1 Objectives Identification (A.2.1)

It is a fact that the client does not always realize the potential of simulation and may not be able to identify certain objectives that may not be directly visible. This is where the modellers' expertise knowledge and experience come into operation. Therefore, it is the duty of the project team to suggest all potential objectives concisely and rank them according to their importance. Such objectives – problem objectives - fulfil the functional requirements directly related to the problem or system. However, setting of objectives to show how the model is reviewed and how the project is planned, taking the time constraint available into account are also essential components for a successful project. Future use of the model for some other purposes is another factor that the project team should concern during this the objective identification task. With this view, the objective identification task delivers the following three task documents into the MRD database. They are;

A.2.1.1 Problem Objectives: with the aim of illustrating the results to be targeted under different conditions. Each objective may consist of:

- Rank,
- Achievement,
- Any measurement, and
- Any constraints.

A.2.1.2 Model Review Objectives: with the aim of highlighting the time scales for experimenting and reviewing the model. The time scales to be considered for:

- Run speed of the model, and
- Visual displays.

A.2.1.3 Work Plan Objectives: with the aim of presenting the target time frame of the project and its various stages. This includes:

- Major activities in the work and review stages,

- Activity time-scale,
- Additional milestones, and
- Personnel involvement.

A.2.1.4 Future Use of the Model: with the aim of recording the potential usage of the model for some other purpose(s) in the future:

- Short-term use, and
- Long-term use.

4.4.2.2 Experiments and Reports Identification (A.2.2)

It is an agreed practice that the experiments are to be articulated as early as the objective setting task in simulation. Although the objectives describe what should be achieved, they may not indicate how they are achieved. The answer to this question may already have been addressed during the problem analysis task as potential solution strategies. Some of the solution strategies may have direct impact on experiments whilst others may not have. Therefore, brainstorming in consultation with all direct participants, to prepare all possible methods of attaining the objectives and then to separate them as excluding and including experimental factors, are vital processes in this task. Also, the reports to be generated from experiments should be decided in this stage to measure the extent to which objectives have been achieved and to highlight the problems that prevent the objects being achieved. Experimental factors may change as the project progresses. However, the time spent on identification of experiments and reports at this stage will not be wasted with the complex simulation models dealt nowadays. Therefore, this task enriches the MRD database with the following two task documents.

A.2.2.1 Experiment Details: with the aim of presenting how the objectives are attained.

Each record may consist of:

- Objective,
- Experimental factor(s),
- Factor range or levels,
- Data entry procedure and
- Any assumptions made.

A.2.2.2 Report Details: with the aim of presenting what and how the results from experiments are presented and evaluated for domain query. This may include:

- Experiment,
- Values to be reported,
- Methods of reporting, and
- Method of viewing.

4.4.3 System Investigation (A.3)

Once the problem is defined, the project team is ready to commence the modelling process. Similar to other cases, a strategic planning is required before commencing a real model building process. Therefore, it is functioned under system investigation stage, which is also known as conceptual modelling. First, a brainstorming session takes place on 'what should be included into the model' and 'how much details should be modelled', considering the entire process. Generally, the client has a tendency to model 'everything', but it is an accepted fact that inclusion of too many system components and too much detail is a waste of time. Similarly, a narrow model with less detail results the model accuracy opened to question. In this respect, problem objectives play a key role in determining the structure of the model as well as in establishing the boundaries of the system and the model scope and level. This is where the experience of the project team play vital role to narrow downs the system according to the problem objectives.

4.4.3.1 Process Analysis (A.3.1)

Generally, the client may presents flowcharts, layout diagrams, schematic displays, written descriptions of all operations, processes, equipments etc. of the existing or proposed system, without evaluating the exact requirements according to the objectives. For example, they may present the entire picture of the plant, but the problem may be only in the area of assembly. Therefore, it is the responsibility of the project team to exclude unnecessary functional areas from the domain query and convince the client the exact requirements related to the objectives. This narrowing down process senses the client to feel much about the direction of the project with their greater understanding about the system (rather than the model development). Thus, it may change the problem objectives identified before. Although this task makes the team to aware about the system components, inclusion of system components to the model is determined in the next task – scope and level. Therefore, it is important, in this stage, to present overall picture of the system and the environment, enabling the document user to differentiate inside and outside of the system boundaries. With this view, this task entails the following task document to the documentation database.

A.3.1.1 System Descriptions: with the aim of presenting system description of the study and boundaries of the system. Record may contain descriptions of:

- Overall system (graphical and/or descriptive)
- Functional areas selected,
- Reasons for selection with any assumptions made.
- Special functional areas omitted, and
- Reasons for omission with any assumptions made.

4.4.3.2 Model Scope and Level Identification (A.3.2)

Having identified the boundaries of the system to be modelled, the analysis should now be taken place in order to identify the preliminary system objects to be included into the model. According to the separation concepts presented in Section 2.4.2.3 for reusable models by Bhuskute et al. (1992), Pratt et al. (1994) and Delen et al. (1996); a simulation model comprises of physical, information and control elements. During the process analysis task, the project team may already have obtained a view about physical elements that are to be included into the model as they are the physically available objects in the real-world system. In modelling, the primary focus of the modeller's interest is the physical extent and characteristics of these components. Interestingly, the common approach to model building begins with these physical objects, based on the physical layout and schematic diagrams of the system. However, many categories of physical objects can be found in a manufacturing system. Therefore, it is worthwhile to analyse them in order to identify generic physical elements and to lay a foundation for reusable and tool-independent model representation.

➤ Generic Physical Objects in Manufacturing System

In general terms, physical objects in any manufacturing system fall into one of the categories: parts, machines, material handling (MH) devices, people and space (Clark, 1996) that are easily identifiable as they are tangible in general (see Fig. 4.1).

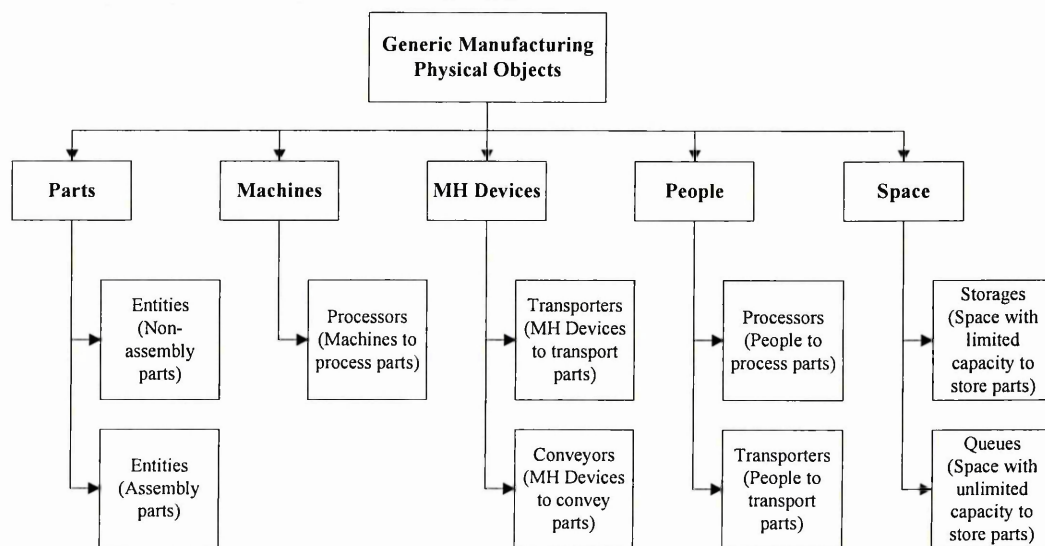


Fig. 4.1 : Generic Physical Objects in Manufacturing

Parts are generally the dynamic objects that arrive into the system; move within the system for obtaining the services from other stationary or non-stationary physical elements, with changing status; and then leave the system after achieving pre-defined requirements. During this dynamic behaviour, they are subjected to processes such as

loading, unloading, machining, inspection, etc. at stationary physical elements (machines or people). They may also be transferred from one station to another by non-stationary material handlers (devices or people). However, analysis of the behaviour of MH devices produces two distinguish material handling methods, namely, transport with devices like AGV, forklifts etc. and convey with overhead trolleys, belts etc. Parts may also be stored in permanent or temporary storages, depending on the individual requirements. In this respect, physical objects in any manufacturing system are categorised in generic terms as in Fig. 4.1.

➤ Physical Elements of a Model

The analysis of generic physical objects and their details show that the physical elements in manufacturing systems are mainly fallen into three categories; entities, resources and queues. Entities are the dynamic elements, like parts, products, components that move through the system while services from other physical elements are being obtained. Resources are the elements such as machines, people, MH devices and tools that provide the services to the entities in many ways like processing, transferring, etc. The limited space, i.e. storage, is used to store the entities and then un-store them whenever another activity or operation is to be performed. Unlimited space, i.e. queue, provides the real or imaginary (logical) waiting space for the entities whose movement through the model has been suspended due to the status of resources, for example, a working machine, over flow of the storage, etc. Analysis of storage objects shows that it can be considered as a queue with a limited capacity. However, it is now considered as a resource object for the purposed of model representation, as a queue may be incurred due to the suspension of entity movement to a storage resource. However, as the queue (unlimited space) is mostly an imaginary object that allows the entities to wait for subsequent operation, it is not considered as a physical element in this context (see Fig. 4.2).

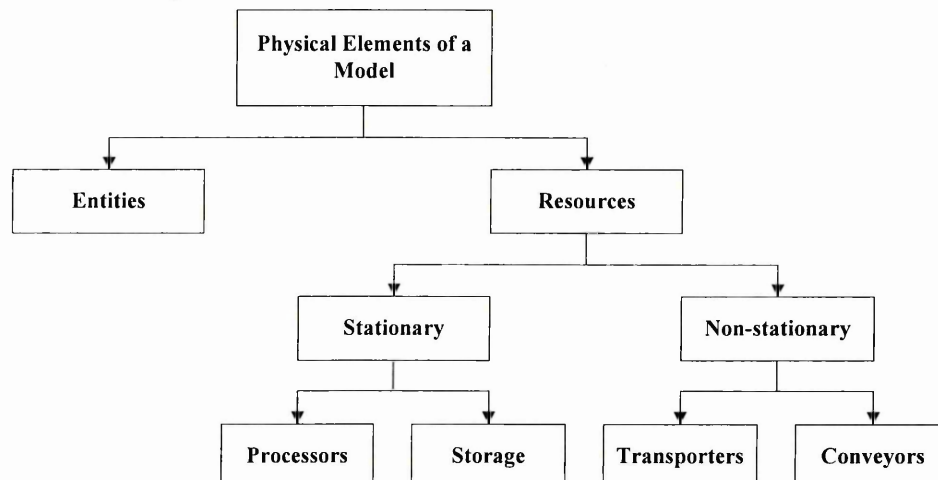


Fig. 4.2 : Physical Elements of a Model

Resources can further be categorised considering the nature of the movement, i.e. moveable or stationary and the services provided to the entities. Accordingly, classification of physical elements can be illustrated as in Fig. 4.2.

Within the system boundaries established in the previous task, physical elements of a model, which are fallen into one of the categories discussed previously, are identified and reviewed for the purpose of deciding whether the element should be included into the model. This process is known as establishing the model scope. However, some of the physical elements may have a major contribution on the problem objectives and results, whereas, others play minor role in the model or may be useful in the future development.

The details of each physical element are then taken into account to determine the level of details required to keeping the model accuracy. In general, there is little difficulty in deciding the model output, if the goal has been defined explicitly. The real difficulty arises in determining which input details produce the desired output. It is a known fact that the analyst tends to include too much detail. Once again, like in model scope, too much detail doesn't necessarily mean more model accuracy. Rule of thumb is that models should include as little details as possible. More details can be gathered at a later stage rather than spending much time for collecting data that has no real impact on the model results. Classification of elements as major and minor is significant in this respect.

Decision on model scope and level of details is usually a trade-off between the accuracy of the model, project cost and time. The greater the degree of detail to be modelled, more precise and expensive input data will be required. Therefore, the model must include only those aspects of the system relevant to the project objectives. Also, this task may overlap with model scope and level reduction task, which is to be performed in the next step. Nevertheless, it is assumed that this task entails with itemizing system physical components with their details without considering model simplification process. In this view, this task delivers the following task document into the MRD database.

A.3.2.1 Physical Elements and Details to be modelled: with the aim of presenting the system physical objects and their details decided to be included into the model.

Each record consists of:

- Types of physical element,
- Name of the physical element.
- Contribution level (major or minor),
- Level of details (Individual operations/set-ups/schedule/breakdowns/etc.),
- Any assumptions made or special reason for selection of elements and details.

4.4.4 Model Formulation (A.4)

It is rather easy to say ‘identify the system elements and their details required for the model’. However, in reality, it is a complex process, particularly with the view of keeping the model as simple as possible. Furthermore, it is not convenient to represent all the characteristics of a real-world element in a simulation model due to, for example, insufficient details. On the other hand, such a detailed-representation may not even be necessary for objectives concerned and leads to over complex model. Therefore, a need arises to simplify the model keeping it with highest level of abstraction. Simplification is generally performed with a reduction of the model scope and level using the commonly available methods such as ‘black box’ approach. However, the simplification inevitably leads to a loss of model accuracy and the modellers need to bear such losses. Even with the simplified model, building and running the model, as a single unit is a time-consuming process. Splitting the model into sub-models and building and running sub-models independently avoids such drawbacks. Hence, sub-models are to be identified for further simplification.

4.4.4.1 Model Scope and Level Reduction (A.4.1)

Having obtained the overview for conceptual model in previous tasks, a possible reduction of model scope and level is now considered, bearing losses in model accuracy, output, visual effects, animation etc. Some view on this respect may already have taken into consideration during the model scope and level identification task. Grouping physical elements is the most common simplification technique. With grouping, a group of elements are represented by a single element when details of each element are not significant. Secondly, one element can also be substituted to another when the available details of substituted element are less. Regardless of what is being represented, the first and the second categories encounter embodying an element and/or details or a group of elements by another element and/or details. Thirdly, some elements or details are excluded when a particular situation arises, occurred infrequently or in order of pattern. In this view, the simplification task centres around the following two task documents to enrich the MRD database. They are;

A.4.1.1 Simplified Elements and Details: with the aim of presenting how the model scope and level were reduced in model simplification. Each record may contain:

- Elements and details to be grouped or substituted,
- Representing element and details,
- Reason and any assumption made in grouping or substitution, and
- Any results that are likely affected.

A.4.1.2 Excluded Elements and Details: with the aim of presenting the objects and details that are likely to have an influence on the model objectives, but that are omitted due to some reasons. Each record may consist of:

- Elements and details to be excluded,
- Reason and any assumption made for the exclusion, and
- Any results that are likely affected.

4.4.4.2 Model Splitting (A.4.2)

A simulation model, particularly for a large system, should be flexible to accommodate easy changes and have a fast model execution during the experimentation. Modelling with such characteristics involves a measurement and a study of individual functional areas as well as the entire system. If the system is considered as a collection of sub-systems, which are represented by sub-models that are independently created, modified, and saved, such an approach allows the model to be built fast, run fast and quick scenario analysis. However, the major challenge is the splitting the model into sub-models to ensure that they could be interfaced to the overall model in order to evaluate the overall system performance under different scenarios. Therefore, the model splitting process begins with isolating subsystems with their elements, keeping the interest on how they are interfaced with the overall model. Lower level of subsystems, if available, are then identified from subsystems that already been separated. Such an attempt lays a foundation for the model-building phase and assures the speedy and quality of the simulation results. However, if subsystems cannot be isolated in a way that makes the system easy to comprehend, then the entire system should be modelled as a single unit. In this respect, the model splitting task delivers the following two task documents into the MRD database. They are;

A.4.2.1 Subsystems: with the aim of presenting details of each subsystem and argument implicated in their isolation and interfacing process. Each record may consist of:

- Name of subsystem,
- Elements, next lower level subsystems, if any, included,
- Inputs to sub-model,
- Outputs from sub-model, and
- Techniques used in formation of sub-systems, and any assumptions made.

A.4.2.2 Model Layout: with the aim of presenting hierarchical relationship of sub-models, respect to the overall model and their details. The record consists of:

- Hierarchical diagram of the model,
- Model level names and IDs, and
- Description of elements in each level.

4.4.5 Data Acquisition (A.5)

Having formulated the model, the input data required for the model are now to be acquired. Data are essential for three purposes: building the model, validating the model, and performing experiments. Any of these stages is not currently functioning. But, the data acquisition is to be commenced at early stage, as it is considered as the most aggravating, challenging and time-consuming aspect in simulation project (Sadowski and Grabau, 2000).

Although the data acquisition is simply interpreted as gathering numbers, it is only one aspect of the situation. The simulation analyst should discover whether data are available, collectable, pertinent, and valid and how data are gathered. Therefore, the data acquisition becomes a complex issue in simulation projects. Having taken the complexity in data acquisition, Robinson (1994) categorises the data into three main types: already available data, not available but collectable data and both not available and not collectable data. Single or multiple sources may be available for data in the first and second categories. Therefore, they are collectable with reasonable effort. The method of dealing with unobtainable data (the third category) is a challenging issue. They are usually estimated. The collection of some data may even be left beyond the end of the project for the model executor to deal with.

4.4.5.1 Data Collection and Estimation (A.5.1)

This primary task basically engages with data identification, data collection and/or estimation and reviewing the accuracy of data. Data required broadly vary with the system and the problem objectives, more specifically, on the model elements. In this respect, the physical elements of the model have already been dealt intensively in previous tasks. But, for any system to be functioned and to obtain the required results, the system should have adequate instructions to perform activities like machining, inspection, and transportation of parts. Those instructions could be in the form of 'information components' like operations, bill of materials of an assembly product, routing etc. They also could be 'control components' like workstation controller, queue controller etc. as suggested by Bhaskute et al. (1992). Therefore, not only the data related to the physical elements, but also data in the information and control elements of the model should now be taken into account. However, a need for an intensive analysis of information and control elements are to be sought in this stage, not only to identify data items, but also for early preparation of reusable model representation task which is due in not so far in the project time scale.

➤ Generic Data Items in Physical Elements

Generic physical elements, discussed previously, are entities, processors, storages, transporters, and conveyors. Common data items of each of these categories are shown in Table 4.1. However, all of them may not be found in a single project or some of the

data items may not be required an attention at this stage of the project. Any simplified element can be treated with the relevant element (e.g. a dummy processor to represent a group of machines). The name of the element (shaded in the table) is not a data item, but it was listed in the table to differentiate similar data items. However, the same identity (name) could be used on the model-building task to maintain the consistency of records throughout the model development.

Generic Physical Element	Data Item	
	In Common Form	In Specific Form
Entity	Entity ID	Name of the entity
	Entity data	Bill of material
Processor	Processor ID	Name of the processor
	Processor data	Number of processor units
	Resource cycle data	Schedule
		Failures
Storage	Storage ID	Name of the processor
	Storage data	Number of storage units
		Storage type (individual/shared)
	Resource cycle data	Schedule
		Failures
Transporter	Transporter ID	Name of the transporter
	Transporter data	Number of transporter units
		Docked position(s)
	Resource cycle data	Schedule
		Failures
Conveyor	Conveyor ID	Name of the conveyor
	Conveyor data	Number of conveyors
		Conveyor segment length
		Number of rows per unit length
	Resource cycle data	Schedule
		Failures

Table 4.1 : Generic Data Items in Physical Elements

➤ Generic Information Objects in Manufacturing System

According to Bhuskute et al. (1992), if the primary focuses of the modeller's interest is on the information content of an object; such an object is referred to as information object. It may, or may not, have a tangible correspondent in the real world system. In other words, these objects collectively model the behaviour of the informational aspects of the system. From broader perspectives, as illustrated in Fig. 4.3, part orders, part operations and part transfers, together with their details, are considered as major information objects of any manufacturing system (Bhuskute et al., 1992).

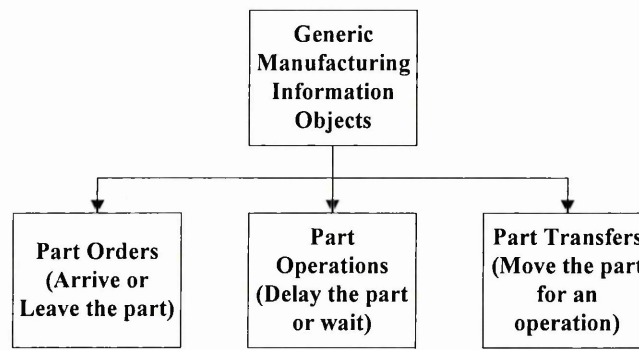


Fig. 4.3 : Generic Information Objects in Manufacturing

In general, the parts (entities) arrive into the system and then they are processed by individual or combined processors sequentially and finally depart the system. During this sequential movement, they may also be transported by transporters and/or conveyed by conveyors from one station to another. Before any activity takes place, the entities may also be stored in a capacitated queue (storage) like a buffer, or need to wait in an infinite capacity queue. With this view, the general activities, which any entity involves in the system, are classified as ‘entity arrival’, ‘entity departure’, ‘process the entity’, ‘store/unstore the entity’, ‘transport the entity’, ‘convey the entity’, and ‘wait for an activity’. These activities can be attached to broader objects, as identified in Fig. 4.3, in order to investigate generic information elements, together with their data items, and subsequently to establish a foundation for software-independent model representation.

- **Part Order:** Entity ‘arrival’ and ‘departure’ are the common activities in any manufacturing system in respect of part order object. Therefore, they become sub-information objects in the part order category.
- **Part Operations:** Part operations associates with services that are obtained from the stationary resources such as processors and storages by entities. The general activities associated in this respect are ‘process’ and ‘store/un-store’ and any type of part operations can be described by these two activities. Operations generally depend upon the types of parts. Hence, each entity has its own part ‘sequence’. The sequence is defined as the flow of parts through the workstations (the stationary resources – processors/storages) in a manufacturing system. The visit ‘station’ may be a single resource (e.g. a machine), or a combined resource (e.g. a machine and an operator) or a member of a ‘resource set’ (e.g. a machine in a group of machines that is assigned for a particular operation). Therefore, part sequence provides an ordered list of stations that the entity visits for its operations. During these sequential operations, a number of parts may be required to group or ‘batch’ together permanently or temporarily before certain operations. It may then be necessary to ‘split’ them back into individual parts, depending on

the subsequent operations. With this view, 'process' and 'store/un-store' objects are considered as major information objects; and 'batch', 'split', 'sequence', 'station' and 'resource set' are the minor information objects in part operations.

- **Part Transfers:** Part transfer activity moves the parts from one workstation to another workstation, which will be a specific station (e.g. a station attached to a resource, a combined resource or a resource set) or a member of a 'station set' (e.g. a station in a group of stations that expect a particular part) using a non-stationary resource. In respect of part transfer, a station may be either physical or imaginary to where the parts move-in or move-out. Station set is a group of stations where parts move-in. The parts may be transferred by a single transporter, a conveyor, or a member a 'transporter set' (e.g. a transporter in a group of transporters expects to move a particular part). Similar to the part operations, the parts may be required to 'batch', prior to transport them and later to 'split' them. With this view, 'transport' and 'convey' objects are considered as major information objects whilst 'batch', 'split', 'station', 'station set' and 'transporter set' are the minor information objects in part transfers.

➤ Information Elements of a Model

The analysis of generic activities, to which any entity is subjected within the system, shows that the primary (major) information elements that drive the manufacturing system fall into six categories: arrival, departure, process, store/un-store, transport and convey as shown in Fig. 4.4.

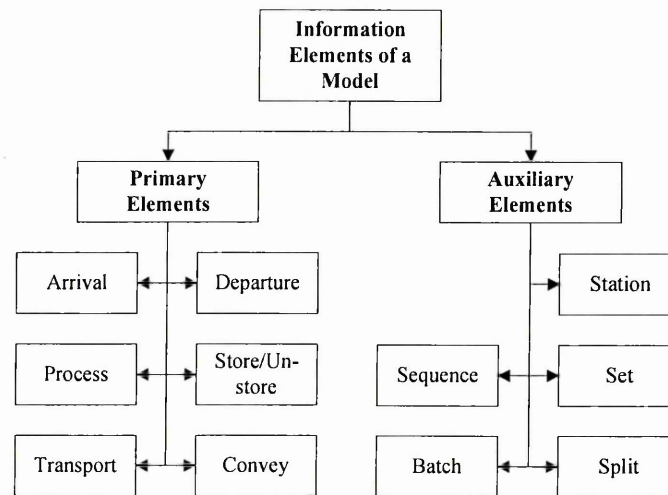


Fig. 4.4 : Information Elements of a Model

However, the other auxiliary (minor) objects such as sequence, station, resource set, station set, transporter set, batch and split are also contribute to the behaviour of the model to great extent. Therefore, they are classified into five generalized categories: batch, split, sequence, set and station. Batch, split and sequence elements are directly

related to the information on activities. One can argue that station and set objects should be considered under physical objects of the model as their characteristics are more biased to the physical elements. But, forming a station element or a set element is solely correlated with activities of a particular entity rather than the physical elements as a whole. Therefore, considering them as information element provides with more flexible environment to build the model in reusable form. A classification of information element as primary and auxiliary elements with their details is depicted in Fig. 4.4.

➤ Generic Data Items in Information Elements

Common data items in primary and auxiliary information elements are shown in Table 4.2 and Table 4.3, respectively. As stated previously, this is an exhaustive list of all data items, which may not be required for a single project. Also, the names of the elements (shaded in the tables) are listed to differentiate the data descriptions of different elements and as a pre-preparation of model representation task.

Generic Primary Information Element	Data Item	
	In Common Form	In Specific Form
Arrival	Arrival ID	Name for arrival
	Arrival data	Number entities per arrival
		Time between arrival
	Arrival limit data	Arrival time
		Maximum number of arrivals
Departure	Departure ID	Name for departure
	Departure data	Name for depart item
Process	Process ID	Name for process
	Process data	Process time
		No. of entities per unit processor
Store/Un-store	Store ID	Name for store/Un-store
	Store data	No. of entities per unit storage
Transport	Transport ID	Name for transport
	Transport data	Transport velocity
		Transport destination(s)
		Distance(s) to destination(s)
		No. of entities per unit transporter
Convey	Convey ID	Name for convey
	Convey data	Convey destination
		Distance to destination
		Maximum rows per unit entity
		Minimum rows per unit entity
		No. of entities per unit processor

Table 4.2 : Generic Data Items in Primary Information Elements

Generic Auxiliary Information Element	Data Item	
	In Common Form	In Specific Form
Station	Station ID	Name for station
Set (Resources/Stations)	Set ID	Name for set
	Set data	Names of members
		Order of members for an activity
Sequence	Sequence ID	Name for sequence
	Sequence data	Step number
		Visit station
Batch	Batch ID	Name for batch
	Batch data	No. of entities per batch
Split	Split ID	Name for split
	Split data	Batch name to Split

Table 4.3 : General Data Items in Auxiliary Information Elements

➤ Generic Control Objects in Manufacturing System

According to Bhuskute et al. (1992), an object can be considered as a control object, if its function is to signal the appropriate action to be taken for different activities in different stations. For example, when an entity has a choice of a machine in a resource set for its operation, the resource selection becomes a control object. The control objects are generally coupled with the physical and/or information elements. For example, selection of a machine from a set for processing the part is simply coupled with processor object, process object or both. However, separating these control objects from other objects provides a soft coded environment to make changes easily to the model. With this view, any object that control the behaviour of the system is considered as 'control' object. The analysis of control behaviour of manufacturing systems indicates that the generic control objects should relate to 'distribution of parts in/out', 'ranking of parts in a queue for the next service', 'selection of a member of a set for a part to get the service' and 'prioritising an activity to obtain a service from a resource'. The criterions for these individual decisions are based on well-established rules. However, in order to sustain any rules that are unable to represent the control behaviour or use defined rules, 'decision' object is necessary for model representation.

➤ Control Elements of a Model

The analysis of generic control objects and their details in a manufacturing system discloses that control elements fall into common 'decision' element and four formalized categories, namely, 'distribution in/out', 'ranking', 'selection' and 'priority'. This classification reflects the most of the characteristics of three categorize (queue controllers, assembly queue controllers and work centre controllers) discussed by

Bhuskute et al. (1992). These elements in conjunction with the physical and information elements provide more natural environment to represent the model independently from the software and in reusable form. The diagrammatic representation of different control elements is shown in Fig. 4.5.

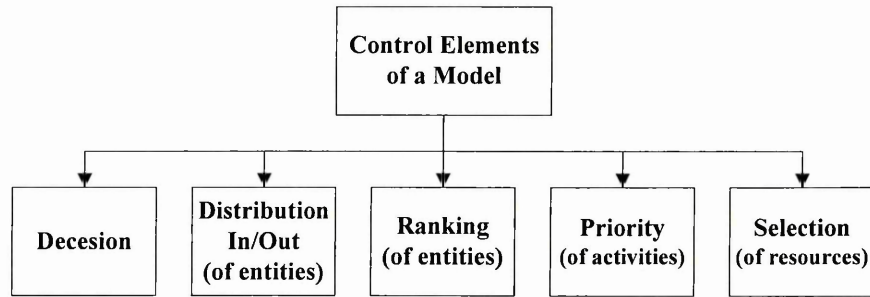


Fig. 4.5 : Control Elements of a Model

➤ Generic Data Items in Control Elements

Common data items of control elements are shown in Table 4.4. The list given in the table is only for illustration of potential data items of control elements. On the other hand, an attention on these items may not be required at this stage, as most of them are easily collectable and are input as model codes or in menu driven items.

Generic Control Element	Data Item	
	In Common Form	In Specific Form
Decision	Decision ID	Name for decision
	Decision data	Decision criteria
Distribution In/Out	Distribution ID	Name for distribution
	Distribution data	Percentages in distribution
Ranking	Ranking ID	Name for Ranking
	Ranking data	Ranking criteria
		Tie breaker
Selection	Selection ID	Name for selection
	Selection data	Selection criteria
Priority	Priority ID	Name for priority
	Priority data	Priority criteria

Table 4.4 : General Data Items in Control Elements

Exhaustive list of generic data items of physical, information and control elements presented previously provides the guidelines to identify the data to be acquired. The data acquisition begins with preparing a list of data items for each element with their purposes and then identifying the data sources. If more than one source exists, the team needs to decide the right source for high level of reliability and accuracy of data. If data is not collectable due to lack of time, unavailability, or any other reason, they need to be estimated. Some data collection may even be left beyond the end of the project, for the model executor to deal with. In worse case scenario, some data needs to be omitted. This

situation may even lead to further simplification of the model scope and level or to changes in problem objectives, if no alteration affects critically on the aim of the project. If all fails, project should be abandon before any time is wasted. The collected or estimated data are then evaluated for their accuracy and validity by some means, as the level of accuracy influences the model performance. However, the accuracy may be an approximate judgment at this stage of the project. As suggested by Robinson (1994), this is best accomplished with group meetings attended by all involved parties.

However, the collected or estimated data may not possess in the same format that particular software expects. Therefore, they are to be formatted according to the software requirements. In this entire process, some assumptions may have to be made about the data, sources, accuracy, or the data acquisition process. Therefore, from documentation point of view, it is essential to present details of data items; how they are collected or estimated with acceptable level of accuracy; and how data are fed into the model. Therefore, this task delivers the following two task documents into the MRD database.

A.5.1.1 Data Description: with the aim of presenting the data required for different types elements and associated details in data acquisition. Each record may contain:

- Types of element,
- Element name,
- Data item,
- Purpose of data,
- Data input format,
- Source(s) used or how it is estimated,
- Any alternative sources or way for estimation, if available,
- Judgement on the level (%) of accuracy of data,
- Any assumptions made on acquisition/accuracy,
- Responsible personal for data acquisition, and
- Any changes expected in data in future.

A.5.1.2 Data Omitted: with the aim of presenting the data items omitted and results that are likely to have an influence due to data omitting. Each record may contain:

- Element type/name,
- Any data items omitted,
- Reasons for omitting and any assumptions made, and
- Any results that are likely affected.

4.4.6 Project Validation (A.6)

All previous steps have been focused on laying a strong foundation for the model building process. In other words, specifications of the project or terms of reference have been determined. However, approval of the project specification from all parties involved in the project are paramount important at this stage, before proceeding to the complex model building process. Communicating the specifications and receiving feedback at early stage give the opportunity for the project team to address any uncovered problems before building the simulation model and to avoid complex and lengthy changes at later stages. If an agreement cannot be reached, the specifications are to be re-issued and comments are to be re-sought iteratively until the specifications are agreed. This process is known as the project validation or testing the validity of the conceptual model. The benefits from project validation lie not only with the model technical details but also with the management information that enables the client to ensure that the work is progressing in right direction and at right level.

The specifications are typically a summary of objectives of the problem and the project work, and the proposed modelling method with brief model technical details. However, the details to be included for specifications would vary with the complexity of the problem as well as the organizational requirements. The general rule of thumb is that it should endow with a briefing of model and project management details so that all personnel contributed towards the project could understand the benefits from the project and the approach to project; and review the status of the project at any time. The specifications may be presented, either in written form or in verbal, or both. However, it is an accepted fact that a combination of written and verbal, perhaps with formal presentation, gives a better feedback and thus to ensure the success of the project validation process. Even after the specifications are validated, it is natural that the changes to specifications may occur at any time in any stage as the project progresses. Overall, the project validation is considered with two primary tasks: project summarization, which provides the details related to the management, and model summarization, which presents technical details of the model. Hence, this sub-phase seems to be a sort of phase-end documentation, for which appropriate task documents could be useful.

4.4.6.1 Project Summarization (A.6.1)

This task engages with presenting the information required for the decision makers, enabling them to understand how the project fulfils their requirements; and with validating those details before proceeding to the model building stage. As they are the recipients of the results of the project, the problem objectives with achievable benefits and how they are achieved within the limited resources and time available are the important specifications

that need to be communicated. Such specifications enable them to aware the direction of the project; review the progress at any stage of the project and aware the individual responsibilities in the project. It is also a fact that all changes to specifications provoked during the validation or future expectations may not be able to handle within the project. Therefore, the handling such changes within the project environment is another issue for the management to deal with, as they influence the time and cost of the project. Alternatively, management may take a decision to handle such changes in the second project. Regardless of the decision, the reasons or justification for the decision are important for the team members to be aware with. Therefore, from documentation point of view, this primary task produces the following two task documents.

A.6.1.1 Validity of Project Specifications: with the aim of presenting the details in validating the specifications relevant to the management. Records may consist of:

- Project specification validated,
- Participants in the validation, and
- Any alterations proposed to project specifications.

A.6.1.2 Dormant Changes in Project Specifications: with the aim of recording suppressed or delayed changes in specifications. The records may contain:

- Proposed changes to project specifications,
- Methods of handling such changes, and
- Any results that are likely affected.

4.4.6.2 Model Summarization (A.6.2)

Although the project summarization provides the management information about the project, those details are not adequate for direct participants in the project such as modellers, model executors. Technical details related to the model are therefore to be presented as model specifications so that the direct participants could visualize the overall picture of the model. Therefore, this task presents the model technical information comprised of the boundaries of the system; the approach to modelling process and data acquisition; etc. Such details enable the model developers and other participants to understand the limitation of the model, and the systematic approach to the model building process. Some of the details, such as problem objectives, may also have to be repeated in this task for effective communication and thus validating the conceptual model effectively. With an appropriate level of agreement to the model specifications, perhaps after re-issuing, an additional time-scale and milestone may be required to establish at this stage for structured and incremental model building process. With this view, this primary task enriches the document database with the following two task documents.

A.6.2.1 Validity of Model Specifications: with the aim of presenting the information in validating process of project specifications, relevant to direct participants in the project. Records may consist of:

- Model specification validated,
- Participants for validation, and
- Any alterations proposed to model specifications.

A.6.2.2 Additional Time-scale and Milestone: with the aim of recording additional milestone, established for structured and incremental model building process. Each record may consist of:

- Sub-models and overall model,
- Responsible personal, and
- Time-scale.

4.5 Task Documents in Phase 02: Model Building and Testing (B)

Once the specifications are agreed or the conceptual model is validated, phase 02 - the model building and testing – of the project can be started to ensure the correct representation of the real-world system. The model building and testing is mostly an incremental and iterative process. Therefore, breaking down of this phase into sub-phases and primary tasks sequentially has become a difficult issue to be addressed. This may have been the reason that many authors have not attempted to split the process into smaller activities. However, for the purpose of task documentation, it is vital to break this phase into smaller tasks, keeping the norms established for creating task documents in attention.

Accordingly, this phase commences with designing and representing the structure of the model to display the flow of entities interacting through different activities of the system. The structured model is then translated or coded into a computer model with respect to the coding method used, i.e. through languages or simulators (mostly). However, model is to be verified to ensure that the program executes as intended. Verification is performed in small steps iteratively, wherever possible, as the coding progresses. Therefore, incremental model development and the verification processes are considered as simultaneous activities. Once the model is coded and verified, it is essential to validate the model to guarantee that a desired level of accuracy between the model and real system is established; and to test the sensitivity of estimated data before beginning the experimentation. Although the above steps in this phase are considered sequentially, they cannot be separated from one to another and their behaviour is highly iterative in nature.

4.5.1 Model Representation (B1)

Planning of the model structure, before model coding or translation, is known as model representation (MR) or structuring the model. In general, model development (model representation, coding and verification) process is very much software dependent. However, instead of restricting to particular software, the model can be represented through neutral graphical representation techniques such as flowcharts, IDEF, ACD, and UML to describe the interactions between different model elements. Such an attempt not only provides the structure of the best modelling method but also software-independent written document of the model. Such documentation lays a foundation for model coding, verification, and validation. Although the model representation generally delivers a static model, whole or part of the model may be translated into the executable model, depending on the structure of the model and the tool used for the model representation.

4.5.1.1 Model Representation for Documentation

Many authors consider model representation as model documentation. Therefore, the model representation has become a key element in this research. In order to substantiate this fact, Robinson (1994) states many benefits from a structured model representation and emphasizes the significance of this step, saying ‘a day on a paper saves a month on a computer’. Despite the benefits, the following negative characteristics in respect of the model representation (or model documentation) are found in the literature.

- Many practitioners opt for direct model coding with chosen software, instead of beginning with model representation (documentation) (Schormann and Perera, 1999).
- Whenever the model representation is attempted, it appears that mostly used common practice is to produce flowcharts, biased to the simulation software being used, to represent some aspects of model logic with varying degrees of details.
- If the model representation and model coding is treated as two distinct activities without direct exchange of details (at least some extent) from former to latter, then model documentation becomes less popular (Benz et al., 2001).
- Designing of a generic methodology for model representation in order to translate the representative model into the executable model become complicated task due to different structure in existing commercial software, as discussed Section 2.3.3.
- It appears that designing of generic methodology for model representation can only be ascertained through software-independent reusable model elements.
- Elements can only be reused for any simulation software, if they facilitate one-to-one function unlike in most procedural nature software as discussed in Section 2.3.3.1. In traditional paradigm, the components tend to perform more than one function (one-to-many-functions) due to hard coding of various details together.

➤ Software-Independent Model Documentation

From the brief discussion in previous section, it appears that the separation of model elements as physical, information and control elements provides the concept for tool independent model representation. The separation concept (Bhuskute et al., 1992; Pratt et al., 1994; Delen et al., 1996) has been proven for tool-independent model representation ('base model') in multi-tool environment such as layout model, queuing model, simulation model, etc. Therefore, the same approach can be applied for simulation software independent representation or documentation. The separation of model elements and their generic and formalized categories together with data items have already been discussed. Once the elements are separated and their generic elements are identified, model representation can be performed incrementally and natural way for each sub-model and for overall model to illustrate the interaction of elements. However, the model represented with physical, information, and control elements is partially described. Therefore, some aspect of logical procedure with data associated with major elements are to be added to compose the model transferable in comparable with the executable model. The context associated with software-independent representative model is shown in Fig. 4.6.

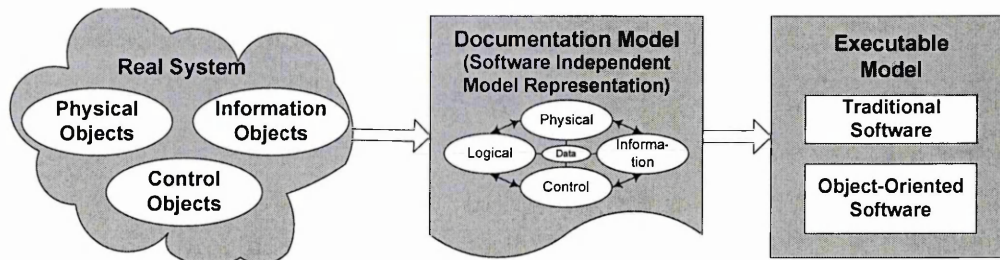


Fig. 4.6 : Software-Independent Model Documentation

➤ Logical Elements

Model representation with physical, information, and control elements illustrates the behaviour of the model. However, some aspects of logical procedure should be incorporated in order to drive the model, before translating the model. For example, an entity needs to 'capture' a processor, whenever the processor is free and available to get the service. Then the entity needs to 'free' the processor when the service is over for other entity to provide the service. Inclusion of these logical elements to the model embeds the logics of the executable model that cannot be described with other elements. Once such logical elements are added, the model is ready for translating to the simulation software through an appropriate interface. A complete list of logical elements cannot be provided, but 'assign' element is considered as a vital element for capturing the model logic and for assigning the data interacting with other elements.

➤ Data Elements

Data in a model associates with physical, information, and control elements. They are mostly the attributes to individual elements. Data in control elements takes a little proportion, compare with other data, and therefore they are not considered as data items. From the model documentation point of view, individual categorize of data should be grouped together and recorded separately. For example, process times for a particular entity or for all entities could be recorded separately. If any element assists to group and record data, such an element is considered as a data element. If data are presented with specific format that allows translating them into the executable model, then it would be an additional benefit from documentation. Therefore, the data elements should assist the document producer to identify the different data items interacting with other elements and then to present them with a specific format for updating the data values. The analysis of generic data items reveals that the following data items can be separated for documentation purpose and to facilitate for model translation:

- ‘Process Times’ as attributes to represent the delay time of a process.
- ‘Resource Cycle’ to represent the schedule and failures of resources.
- ‘Transport Velocity’ as attributes to represent the velocity of transporter.
- ‘Transport Distance’ to capture the transport distances in matrix form.
- ‘Convey Distance’ to capture the convey distances in matrix form.
- ‘Convey Min/Max Row Units’ as attributes to represent the min/max rows occupied per unit entity.
- ‘Other Assign Attributes’ to capture other attributes assisted for capturing logics.
- ‘Expressions’ and ‘Variable’ to describe the variable assigned to the model.

The above list is not a comprehensive list and some of them may not be direct data items. However, treating them as data items enables the user to record them separately for documentation and also to build the model (or document) without interference of model data. Later, the data tables can be updated or someone else can be directed to prepare data table. Moreover, it keeps confidentiality of data, in case of model reuse, as it makes provision to remove them without affecting model logics.

The above description portrays that the model representation is seen in two perspectives: structuring the model and capturing the data, but, in reality, these two activities cannot be separated. Nevertheless, from documentation point of view, such a separation is essential to record the details separately. Such features can be achieved with software-independent reusable model elements, discussed previously. In this respect, data elements play a vital role to capture model data as a separate activity. Therefore, this sub-phase is divided into two primary tasks: model structuring and data capturing.

4.5.1.2 Model Structuring (B.1.1)

The most appropriate way of structuring the model is with the use of graphical symbols, defined for each model elements that are separated to perform on-to-one function. All elements jointly represent the logics of the executable model by graphical means, but supported with some textual presentation to differentiate the various elements in the same type. If the elements used to represent the model is software-independent and reusable; and the techniques adopted to describe the model is neutral technique, then the model represented can be considered as generic, and makes a provision to translate it into software-dependent executable model. Such a model, which is developed for documentation and translation purposes, is referred to as 'Documentation Model (DM)'. With this view, the model-structuring task delivers the following task document into the MRD database.

B.1.1.1 Documentation Model: with the aim of presenting the diagrammatic structure of the model so that the same structure could be translated into the executable model. The contents of the documentation model cannot be stated without a specific example. Nevertheless, it generally consists of:

- Physical elements, representing the system layout and entity flow.
- Primary information elements, describing the process flow of entities and establishing the relationship with physical elements.
- Auxiliary information elements, assisting to describe the entity flow.
- Control elements, illustrating the control behaviour of the model.
- Logical elements, capturing model logic and making relationship with data.
- Data elements, indicating the data required in different levels of the model and preparing to produce data table in the next task.

4.5.1.3 Data Capturing (B.1.2)

Generally, data are captured while the model is being built. Similar way, with the help of the data elements in the documentation model, the individual categorize of data can be captured with according to the specified format that facilitates model translation. Although the use of same format for recording model data gives unnecessary details, such approach would be an additional benefit from documentation process. With this view, the data-capturing task delivers the following task document into the MRD database;

B.1.2.1 Model Data: with the aim of recording the different set of data associated with the model. Although, contents vary with category, they, in general, may have:

- Associated element type and name,
- ID or expression and types for data items, and
- Data values and units

4.5.2 Model Coding and Verification (B.2)

Once the model structure is finalized or the model is documented, the modeller begins the model coding process incrementally with deep understanding acquired during model representation stage. It is an accepted fact that the best practice is to start the model building with a simple model and then continue towards the greater complexity by maintaining the incremental model development. At the same time, verification is performed to ensure that the model elements behave in the manner intended by the model code. For example, if there is a clear close boundary, which may form a sub-model, that region (sub-model) is coded and verified separately. If clear boundaries do not exist, then the modeller may decide imaginary boundaries, considering the verification procedure. If all fails, complete model should be coded and verified at once. This reveals that coding and verification are highly iterative processes. Therefore, breaking this stage into sequential primary tasks becomes much more difficult. However, for recording purpose, this stage is divided into two primary tasks: model coding and documentation, and model verification.

4.5.2.1 Model Coding and Documentation (B.2.1)

Model coding is exclusively software-dependent. Therefore, it may be associated with procedural programming like in general-purpose languages or visual interactive modelling (VIM) in which the programme is generated automatically or both. This implies that the whole coding process is totally led by the rules governed and the facilities provided by the individual software being utilized. While the model is being coded, model codes are to be documented for future reference. Such documentation may not be necessary, if the model representation (model documentation) stage has been completed comprehensively and is updated to capture the alterations made during the coding. Even if the model representation is performed well, some attention on documenting the model codes would still be worthwhile for future references. Nevertheless, from documentation point of view, any method cannot be proposed for documenting the model codes due to existence of various approaches in commercial simulation software. However, the following guidelines are suggested to make the documentation of executable model self-explanatory.

- Use meaningful names for elements, attributes, variables, and expressions.
- Avoid employing algebraic symbols like numeric figures, alphabetic letters individually.
- Use any facilities provided by the software for documentation intensively to explain the important logical steps and variables. Some software provides documentation facilities such as additional memos, notepad, text files, summaries, etc.
- Correlate the model, in case of VIM, with physical system for easy understanding.

Some of the above guidelines are true even for the model representation task. If the documentation model is represented with such guidelines, then the consistency could be maintained throughout the project life cycle. Such maintenance may not be necessary, if the documentation model could be converted or translated into the executable model, at least partially, making the documentation worthwhile.

Regardless of the way the model is coded (i.e. direct coding or model translation), the task deliverables in this stage could be hard copies of the model (symbolic/text/both) with other associated text and graphical records facilitated by the software being used. With this view, the following task document is proposed for referencing with the executable model.

B.2.1.1 Model Details: with the aim of presenting the model code and other associated details for illustration purpose. The record may consist of:

- Software used,
- Model and other supplementary file names,
- Model codes (symbolic/text) in printed form, and
- Other supplementary details, if available.

4.5.2.2 Model Verification (B.2.2)

Model verification is the process of ensuring that the model (rather model elements) operates or executes as intended and the model elements behave as the real-world elements. The first point indicates the need for checking model code whilst the second designates the necessity of running the model for verification. Code verification normally entails detailed desk checking and software testing of codes. This process is known as static model verification. In contrast, in dynamic model verification, the behaviour of model elements is guaranteed. For this purpose, the model or a part of the model is executed under different conditions. The results obtained from tests are used to determine the correct operations of the model elements and code implementation (Sargent, 2000). For both static and dynamic testing, there is no specific rule that a particular technique is exclusively used for a specific project. Therefore, it is the choice for the project team to decide what to verify and how to verify. Hence, the answers to those questions become task details in the task deliverable in the model verification task. However, the proposed records may display only the successful or last resort in this highly iterative task, unless the details are accumulated deliberately.

B.2.2.1 Verification Details: with the aim of presenting how the model verification is performed. The record consists of accumulation of the following details:

- Elements and data verified,
- Technique(s) employed for verification, and
- Brief descriptions on verification process and output.

4.5.3 Model Validation (B.3)

Model validation is generally defined as the process of ensuring that the model represents reality. In other words, it is the process of establishing the desired accuracy between the simulation model and the real system. However, in the simulation literature, there are numerous views regarding the purpose of validation and the stage of validation, but, from the documentation point of view in this research, two validation stages were suggested. The first stage is at the end of the phase one, where the project specifications are validated before commencing the model building and testing stage. In second stage, the accuracy of the coded model needs to be tested to ensure that the model can meet the purpose defined for the project. If the model is used for some other purpose other than defined, it needs to be validated for that purpose separately.

At this instance, it raises a question ‘what function is performed in the model verification in contrast to the model validation?’ The answer is bit ambiguous. Unlike in model validation, in which the overall model accuracy and the ability to meet the objectives are tested, model verification primarily tests the elements individually. However, the analysis of dynamic verification process shows that the verification and validation cannot be considered as disconnected activities and there are much overlapping areas. This is further confirmed by common techniques employable for verification and validation purposes. This implies that both validation and verification (particularly in dynamic verification) could be performed simultaneously. Nevertheless, for the purpose of documentation, it is assumed that the model validation is performed only when the complete coded model is available. Verification can be carried out either or both during the model development and validation. However, in the latter case, recording of verification details entails with updating the previous task document – verification details. In review of the literature on validation (including the verification), it reveals that the model validation process can be seen from two perspectives: validation of model operation and validation of model parameters. Although the process of operation validation cannot exclude the data associated with the model, a special attention is given on inaccurate or estimated data during the documentation in parameter validation task.

4.5.3.1 Operation Validation (B.3.1)

The process of operation validation is especially concerned with determining that the model’s output behaviour has the accuracy required for the model’s intended purpose over the domain of its intended applicability. Therefore, the operational validity becomes a major test in model validation process. This is where the most of the subjective or objective validation techniques and evaluation processes take place (Balci, 1997). The choice of the technique depends upon its appropriateness and the preferences of the project team. Similar

to verification, no method should be used in isolation and it may require using a number of validation checks for complete validation. Regardless of the technique(s) used, if the model is found invalid or not up to the level of accuracy expected due to some deficiency, then the reasons for deficiencies should be investigated and the model should be updated, recommencing from relevant stages until the validity is achieved. Deficiency may arise due to inadequacy of conceptual model, improper coding, invalid data (this will consider in the next task) etc. However, it is to be noted that the proposed recording method provides only the last successful details (the details of the validation accepted) in this highly iterative task.

B.3.1.1 Details in Operational Validity: with the aim of presenting how and what extent the operational validity is obtained. The record consists of:

- Technique employed for validation,
- Process and data description in validation, and
- Results obtained and conclusion.

4.5.3.2 Parameter Validation: Sensitivity Analysis (B.3.2)

During the process of operation validation, it is certain that the data has already been validated, considering the entire model. But, a special attention should be given on the estimated data and the inaccurate data (in data acquisition task, inaccurate data has been identified). Therefore, it is important to determine the effect of any inaccuracies of those data by performing sensitivity analysis (Robinson, 1994 and Kleijnen et al., 2001). Hence, such analysis is considered as subsequent task to operation validation for documentation purpose. Therefore, the details comprised of in this validation process are recorded separately from other validation details to ensure the confidence of the working model or to highlight the results that are likely to affect, in case of failure to prove the accuracy of data.

B.3.2.1 Details in Parameter Validity (Sensitivity Analysis): with the aim of presenting how the inaccurate or estimated data are validated and what impact they impose on the model output. The record consists of accumulation of the following details:

- Data item,
- Brief description on sensitive analysis process, and
- Results obtained, conclusion and any results that are likely to affect.

4.6 Task Documents in Phase 03: Experimentation (C)

Having built, verified, and validated the simulation model; the project team is now in a position to use it to draw inference about the real system. For this purpose, the proposed methods of achieving the objectives are tested and the results are analysed. To perform this, the experiments are designed to answer what needs are to be tested, what data are to be obtained, in what form data are to be obtained, how many experiments are to be conducted,

etc. Once the answers are found, the simulation model is run under different combinations of conditions, designed to obtain different sets of results for the key performance of measures. The results are then analysed to draw inferences on the relationship between the controllable variables and the measured performance. Although it seems that the results analysis is performed on completion of results, it is important to analyse the results throughout the experimental run to identify further experiments for meeting the problem objectives. Therefore, simulation run and reports analysis is considered simultaneously.

4.6.1 Design of Experiments (C.1)

Experimental design is the process of formulating strategies, procedures, and tests for analysing and comparing alternatives to fulfil the goal of the study. It aims to maximize the usefulness of the information produced from simulation runs by minimizing the time, cost, and effort for experimentation. In border context, such a design includes questions of: what the overall purpose of the project is, what the output performance should be, how the random numbers are used for different set of outputs, how to measure the outputs against different set of inputs, how to search optimal configuration, etc. (Kelton, 2000). In finding the answers, the specific questions related to them would arise. They can be considered with two experimental plans, namely, design of strategic experimental plan and design of tactical experimental plan. In former, appropriate experiments in different scenarios are mapped onto chosen key measures of performance. In latter, the detailed experimental conditions of each individual simulation run are determined (Benjamin et al., 2000).

4.6.1.1 Design of Strategic Experimental Plan (C.1.1)

Designing a strategic experimental plan refers to the simultaneous process of deciding upon the metrics that evaluate the performance of the simulation model; and designing of instrumentation to generate the data needed for performance metrics evaluation. The performance measures of the simulation model often do not openly give the answers to the query posed by the domain expert. However, the purpose of building the simulation model in the first place was to provide the sources of information required to answer the domain query. Therefore, the query needs to be mapped onto the performance metrics to be generated from the simulation model. This has already been dealt in the experiments and reports identification sub-phase. However, the reality is that the goal and the experiments may change as the project progresses as a result of previous experiments conducted, if any. Moreover, the decisions taken earlier about the reports - performance measures - to be generated may not have based on the output facilitated by the software being used. Therefore, determining right experiments at right level with appropriate key performance measures to draw valid inferences should still be considered critically.

Experimentation process is generally lengthier and tedious process. Therefore, it is important to design the process, taking the various types of simulation model and their associated characteristics, such as experiment terminating situation, simulation status, and methods of experimentation, into account initially. This gives rough estimation on the time and effort required for both designing and running of experimentation task.

Once the initial investigation is finished, the experiments previously identified are further explored in order to determine the relative importance of each experiment and their factor levels. Then finalized experiments are mapped onto the performance metrics facilitated by the software and additional ones, if required. The outcomes from this task may, therefore, update the experiment and report details set-up at the beginning of the project. Hence, the strategic plan task delivers the following task documents into the document database.

C.1.1.1 Overview on Experimentation: with the aim of presenting overall view of the experimentation process enabling the document user to aware how the experimentation process is designed and conducted; and to what extent the time and effort is required. The record may consist of the following details:

- Model type,
- Simulation terminating situation(s),
- Simulation status, and
- Approach(es) to experimentation.

4.6.1.2 Design of Tactical Experimental Plan (C.1.2)

The tactical experiment plan refers to those activities, which determine the detailed experiment specifications of each individual simulation run. The major issues to be addressed at this stage include in determining of:

- the starting conditions of the simulation run,
- the length of the warm-up period,
- the length of each simulation run, and
- number of independent simulation runs.

Starting conditions may be an alternative to the warm-up period or vice-versa. In both instances, the simulation model is attempted to put into a real state at the beginning of the model run. Although they seem to be alternatives approaches, in practice, they are mixed together to represent the normal working condition (Robinson, 1994). Estimating the warm-up period or the determining the data for establishing the initial conditions generally depends on the characteristics of the model. Once the steady state is reached with the established warm-up period and/or the starting conditions, the model needs to run for collecting results. This now brings issues in determining the length of the run and a number

of replications for each experiment. Values for these factors are influenced by many factors such as model type, terminating point, experimentation approach, etc. reviewed in the previous task. The values are theoretically determined by analysing the results from the model run with graphical techniques.

C.1.2.1 Warm-up Period (WUP) and Starting Conditions: with the aim of presenting details in establishing starting conditions, if necessary, and in estimating theoretical WUP. The record consists of the following details:

- Starting condition established,
- Data and description in establishing starting conditions,
- Theoretical value of WUP, and
- Data and description in the process of estimating WUP.

C.1.2.2 Run Length: with the aim of delivering how run length was determined for each experiment. The record consists of the following details:

- Approximate values and events for run length,
- Theoretical value and confidence of run length, and,
- Data and description in the process of determining the run length.

C.1.2.3 Number of Replications: with the aim of presenting how the number of replications is decided for each experiment. The record consists of:

- Theoretical value and confidence of number of replications, and
- Data and description in the process of determining number of replications.

4.6.2 Experiment Run and Results Analysis (C.2)

During the previous task, critical analysis on what and how experiments are performed and results are obtained were considered. Now, it is the project team (or the model executor) to put them into practice to obtain the results and analyse the results for drawing the conclusions to ensure that the problem objectives are achieved. Despite the fact that the aforesaid tasks are listed separately, in reality, they and the tasks in this sub-phase are by no means separate tasks and are performed with continuous reviewing. Also, there is a gap between the theoretical aspect considered previously and practical experimentation process due to constraints such as time and resources. These factors come into operation during the experiment run task and their effect is evidence during the report analysis task.

4.6.2.1 Experiment Run (C.2.1)

Experiment run is the task of running the model for collecting or observing the result, depending upon the experimentation strategy adopted. In general, this task is one of the time-consuming and tedious processes. It is unusual, if the same experiment is repeated several times or preliminary experiment is conducted until the purpose is achieved. Such

approach may be obstructed by real-world factors such as project time, resources and facilities in the software being used. In such a situation, the team needs to alter the designed experimental parameters and experiment factors with their levels to suit the prevailing situation. Therefore, brief awareness of those practiced details in experiment run not only shows the gap between the theoretical and practical aspects in the experimentation process, but also helps the document user to analyse the reports from the experimentation, accommodating the alterations made to suit the situation. Therefore, the experiment run task delivers the following task documents into the MRD database.

C.2.1.1 Summary of Experiment Run: with the aim of presenting how the experiments are conducted. The record consists of the following details:

- Experiment input description,
- Practiced experimental parameters, and
- Useful notes on practiced values.

4.6.2.2 Results Analysis (C.2.2)

Results analysis task refers to the detailed analysis of model output leading to the generation of information for decision-making. This generally bridges the model building and decision-making process with the aim of ensuring whether the problem objectives have been met and/or the extent to which the objectives have been achieved. The results analysis generally involves variety of activities such as formulating appropriate output matrices, identifying and quantifying output correlation, statistical estimation and initialisation bias elimination (Benjamin, et al., 2000). The procedure to be adopted for results analysis varies with the results obtained, but, in general, the results can be analysed with reference to point estimates and/or to measures of spreads. In point estimates, key performances are presented in terms of mean, median, etc. In measures of spreads, variations of performances are delivered in terms of standard deviation, minimum & maxim values, histograms, etc. Most of the existing software facilitates for such analysis or make provisions for third party software such as spreadsheet to perform such analysis. From documentation point of view, it is impossible or not important to record all details of such analysis. Instead, a systematic recording is required to present the summary of different sets of experiments conducted and sets of results obtained. However, a provision can be made to provide additional details.

Once the results are analysed then the conclusions are drawn, based on the results obtained, and recommendations are made, based on the conclusions. It appears that the conclusions and recommendations are inter-related. Therefore, for recoding purpose, the details in both activities are considered in a single task document. Accordingly, the results analysis task delivers the following two task deliverables into the documentation database.

C.2.2.1 Summary of Results: with the aim of presenting significant results in respect of different experiments conducted. The record consists of the following details:

- Different experiment factors and their values in each experiment,
- Replication number or range of replications,
- Individual or average values in each performance measure, and
- Significant individual results or details in the analysis process.

C.2.2.2 Conclusions and Recommendations: with the aim of presenting conclusions and recommendation & their associated details. The record consists of:

- Conclusions and based details,
- Recommendations and based details, and
- Other details influenced for recommendations.

4.7 Task Documents in Phase 04: Project Implementation (D)

Project implementation is generally considered as the process of placing the recommendations into practice and of documenting the simulation model for future use. However, in the context of this research, the documentation is an integrated process throughout the project life cycle. Therefore, the documentation has not become a distinct activity in this phase. Instead, this phase is concluded by presenting accumulated task documents for different audiences for different purposes. Similar to the first phase, different authors have presented different breakdowns and perspectives for this phase. It has also given a less prominence, compared with other three phases. Robinson (1994) argues that the study may fall into relative shadows through lack of attention to this final phase. This is further enriched by Shannon (1998) saying that the modellers throw the results at the last moment without paying much interest to this implementation phase. Such situation inevitably makes the client with less confidence about the outcomes, though the significant results are obtained. The best way of developing the confidence of the client and other interested parties is through proper communication. Such communication ensures that the recommendations are agreed and project is reviewed. Finally, the project is completed by developing an implementation plan and preparing documents for managing personnel turnover and for future use of the model.

4.7.1 Results Communication (D.1)

Upon the completion of conclusions and recommendations, the client is enthusiastic to implement the findings and would like to see the payback immediately. But, this mainly a matter of the client's confidence about the work, which depends on not only the technical contents of the work, but also how the work is delivered. There may have been considerable success and benefits from the project, but the perception of the client may be somewhat

different unless the results are properly communicated. With proper communication, the project results including the recommendations are reviewed to ensure that the problem objectives are met. This time is also appropriate to review the success or failures of the whole project, as the implementation may take months or years, depending on the environment of the project, to complete.

4.7.1.1 Reviewing the Results (D.1.1)

The most important purpose of this task is to ensure that the results and conclusions are understood and the recommendations are agreed before moving to the project implementation. For this purpose, the results along with the project summary are presented for different participants and seek for feedback to agree the recommendations proposed or to bring new ideas. Such an agreement not only improves the success of the present simulation work but also increases the chances for extending the same project for some other purposes or for a completely new work. With this view, this task delivers the following task document into the documentation database.

D.1.1.1 Review of the Results: with the aim of presenting details in reviewing the results and its outcomes. The record may consist of:

- Method of communication,
- Participants in the review,
- Details presented/reported,
- Level of agreement, and
- Any alternatives to recommendations.

4.7.1.2 Reviewing the Project (D.1.2)

The major purpose of the project review task is to assess the project and to disseminate how the project approach could be improved (Robinson, 1994), at least in future, in reuse of the same model or in a similar project. At this stage of the project, the implementation may have not yet even commenced to measure the success of the project. However, as the time taken for implementation varies from one project to another and participation of simulation team for the implementation is uncertain; this is the most appropriate time to review the project. The project review task can be seen mainly in two perspectives: the project team's viewpoint and the client's viewpoint. The members of the project team have the opportunity to evaluate by themselves on what they have done or what they should have done better. The client has the similar opportunity to assess the project (and, thus, the project team) on what it has achieved and what it should have been achieved. Such a review would benefit not only for project team to sell their performance, but also for others to gather learning experience, particularly in case of project implementation and model reuse. With this view, this task delivers the following task document into the MRD database.

D.1.2.1 Review of the Project: with the aim of delivering the success, failure or further improvement required for the project. The record consists of:

- Status of the problem objectives,
- Perspectives of the project team, and
- Perspectives of the client.

4.7.2 Project Completion (D.2)

Upon the completion of the review stages, the project has now reached to final stage – project completion. Project completion generally involves with employing the agreed recommendations into practice (implementation) and documenting the simulation model for the future use. Project implementation may take shorter or longer time-span, depending on the requirements of the project. Therefore, full participation of the simulation project team may be uncertain for the implementation stage. Regardless of the participation of project team members, a plan should be developed for implementing the agreed recommendations successfully and put it into practice. While the implementation plan is being executed, the final documents could be prepared, by accumulating previous task documents, for different purposes of different audiences.

4.7.2.1 Development of Implementation Plan (D.2.1)

Simulation project can only be considered as effective, if the recommendations are implemented successfully. Implementation may take months or years, depending on the nature of recommendations. Hence, implementation can be treated as a separate project, comprising of a separate implementation team. Also, the implementation depends on other factors, outside the project team's control, for instance, availability of adequate finance and continuous changes in the real world (system). Therefore, at this stage, it is the responsibility of the simulation project team to establish necessary guidelines for implementing and for monitoring stages, in consultation with the client and the relevant parties. This generally comprises of: what recommendations will be implemented, who will implement them, in what stages the recommendations will be implemented, how the success of the implementation could be monitored and how necessary changes, if required, will be handled. With this view, this task delivers the following task document into MRD database.

D.2.1.1 Implementation Plan: with the aim of presenting how the project implementation is performed. The record consists of:

- Recommendations to be practiced,
- Implementation team,
- Implementation stages, if any, and time scale, and
- Monitoring stages and due dates.

4.7.2.2 Completion of Documents (D.2.2)

Authors such as Banks (2000), Law & Kelton (1991), and Shannon (1998) consider the documentation as one of the ending activities in a simulation project. Different sets of details are recommended for different documentation purposes. However, in the context of this research, the documentation is performed throughout the project life cycle as an integral part of the project. The details of each task are recorded in different task documents progressively, not retrospectively. Therefore, the general documentation activity in simulation has become ‘documents completion’. In this respect, it is now the responsibility of the simulation team to accumulate those task documents, which have already been prepared and present them for the future use or for any other purposes.

Such approach brings up the issues ‘What is the purpose of each document that needs to be prepared?’, ‘Who benefits from those documents?’, ‘What task documents are to be included in different documents?’ These issues turn the research process back into the beginning, where similar sets of questions were raised. However, they are now with specific terms that enable the document writer to address them precisely. This situation highlights the necessity for analysing the requirements of different sets of documents to be produced in respect of three variables – the audiences that are benefited from documentation, the purposes of documentation, and the task documents. Although the task at this stage is to discuss the documents that need to be prepared for future usage, it is now more appropriate to capture a broad analysis of task documents in respect of MRD purposes and audiences.

➤ Purposes of Documentation of Different Audiences

First, different purposes in documentation, as identified in the definition of MRD, and against the audiences are analysed in order to identify how different audiences are benefited from the proposed MRD. The results of the analysis are shown in Table 4.5 and purposes are numbered for convenience of the next analysis (Tables 4.6 and 4.7).

Audiences Benefit from Documentation Purpose of Documentation	Model User (Owner)								Model Re-User		Others Interested	
	The Client	Team/Project Leader	The Modeler	System Supporter	Modeling Supporter	Data Provider	Model Executor	Interested In-house Member	Client/Top-Level Manager	Analysar/Modeler	Researcher/Student	-----
1. Dissemination	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
2. Quality	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
3. Maintenance	✓	✓	✓	✓	✓	✓	✓	✓	x	x	x	x
4. Communication	✓	✓	✓	✓	✓	✓	✓	✓	x	x	x	x
5. Motivation	✓	✓	✓	✓	✓	✓	✓	x	x	x	x	x
6. Reporting	✓	✓	✓	✓	✓	✓	✓	x	x	x	x	x
7. Contingency	✓	✓	✓	✓	✓	✓	✓	x	x	x	x	x
8. Enhancement	x	x	x	x	x	x	x	x	✓	✓	✓	✓

Table 4.5 : Purposes of Documentation of Different Audiences

The above analysis displays that MRD is for different audiences for different purposes. For example, MRD is useful for dissemination of accumulated knowledge and for ensuring the quality and creditability of the model for all types of audiences. Whereas, for model re-users and the other interested parties (students, researches, etc.), it is important for enhancing the model reuse. In other words, individual requirements of the different audiences vary with the documentation purposes, as appeared in the table.

➤ **Task Documents for Different Purposes**

The analysis of documentation purposes for different audiences as in Table 4.5 does not show which task deliverable fulfils which purpose. Some task documents may not be significant for all purposes (e.g. the task document, 'form of review' is not required for enhancing the model reuse) whereas some others may be required for all MRD purposes (e.g. 'review outcomes' is significant for all purposes, including the enhancement of model reuse). This highlights the necessity of analysing the different task documents for different purposes. The outputs from this analysis are depicted in the Table 4.6.

➤ **Task Documents for Different Audiences**

The analysis of task documents for different purposes does not show the variation in task documents for different audiences for different purposes. Some task documents may not be important for certain types of audiences, but they seem to be significant for a particular purpose, in general sense. For example, Table 4.6 points out that the 'review outcomes' is an important task document for reporting purpose in general terms. However, it is not be significant for the data provider or the model executor for reporting purpose. Nevertheless, it is vital for the client for reporting purpose. This highlights the necessity of analysing the requirements of different task documents for different purposes as well as different audiences. The outputs from the analysis in respect of purposes, audiences, and task documents are shown in the Table 4.7. For convenience of the analysis, the purposes of documentation are numbered (1-8). This analysis demonstrates that;

- All task documents are significant for dissemination, quality, and maintenance purposes, irrespective of the audiences.
- All task documents fulfil the communication and motivation purposes in all types of audiences in the model user category.
- Reporting and contingencies in model users categorize are fulfilled by various task documents.
- Enhancement of model reuse is also supported by various task documents.

TD ID	Task Document (TD) Name	Purpose of MRD							
		1. Dissemination	2. Quality	3. Maintenance	4. Communication	5. Motivation	6. Reporting	7. Contingency	8. Enhancement
A.1.1.1	Overview of the Client	✓	✓	✓	✓	✓	✓	✓	✓
A.1.1.2	Problem & Need Statement	✓	✓	✓	✓	✓	✓	✓	✓
A.1.1.3	Project Team	✓	✓	✓	✓	✓	✓	✓	X
A.1.2.1	Form of Review	✓	✓	✓	✓	✓	✓	✓	X
A.1.2.2	Review Outcomes	✓	✓	✓	✓	✓	✓	✓	✓
A.2.1.1	Problem Objectives	✓	✓	✓	✓	✓	✓	✓	✓
A.2.1.2	Model review Objectives	✓	✓	✓	✓	✓	✓	✓	X
A.2.1.3	Work Plan Objectives	✓	✓	✓	✓	✓	✓	✓	X
A.2.1.4	Future Use of the Model	✓	✓	✓	✓	✓	✓	✓	✓
A.2.2.1	Experiment Details	✓	✓	✓	✓	✓	✓	✓	✓
A.2.2.2	Report Details	✓	✓	✓	✓	✓	✓	✓	✓
A.3.1.1	System Description	✓	✓	✓	✓	✓	✓	✓	✓
A.3.2.1	Model Physical Element&Details	✓	✓	✓	✓	✓	✓	✓	X
A.4.1.1	Simplified Model Physical E&D	✓	✓	✓	✓	✓	✓	✓	X
A.4.1.2	Excluded Model Physical E&D	✓	✓	✓	✓	✓	✓	✓	X
A.4.2.1	Subsystems	✓	✓	✓	✓	✓	✓	✓	✓
A.4.2.2	Model Layout	✓	✓	✓	✓	✓	✓	✓	✓
A.5.1.1	Data Description	✓	✓	✓	✓	✓	✓	✓	X
A.5.1.2	Data Omitted	✓	✓	✓	✓	✓	✓	✓	X
A.6.1.1	Validity of Project Specifications	✓	✓	✓	✓	✓	✓	✓	X
A.6.1.2	Dormant Changes in Project Spec	✓	✓	✓	✓	✓	✓	✓	X
A.6.2.1	Validity of Model Specifications	✓	✓	✓	✓	✓	✓	✓	✓
A.6.2.2	Additional TimeScale&Milestone	✓	✓	✓	✓	✓	✓	✓	X
B.1.1.1	Documentation Model	✓	✓	✓	✓	✓	✓	✓	✓
B.1.1.2	Model Data	✓	✓	✓	✓	✓	✓	✓	X
B.2.1.1	Model Details	✓	✓	✓	✓	✓	✓	✓	✓
B.2.2.1	Verification Details	✓	✓	✓	✓	✓	✓	✓	✓
B.3.1.1	Details in Operational Validity	✓	✓	✓	✓	✓	✓	✓	✓
B.3.2.1	Details in Parameter Validity	✓	✓	✓	✓	✓	✓	✓	✓
C.1.1.1	Overview of Experimentations	✓	✓	✓	✓	✓	✓	✓	X
C.1.2.1	WUP and Starting Conditions	✓	✓	✓	✓	✓	✓	✓	X
C.1.2.2	Run Length	✓	✓	✓	✓	✓	✓	✓	X
C.1.2.3	No. of Replications	✓	✓	✓	✓	✓	✓	✓	X
C.2.1.1	Summary of Experiment Run	✓	✓	✓	✓	✓	✓	✓	✓
C.2.2.1	Summary of Results	✓	✓	✓	✓	✓	✓	✓	✓
C.2.2.2	Conclusions & Recommendations	✓	✓	✓	✓	✓	✓	✓	✓
D.1.1.1	Review of the Results	✓	✓	✓	✓	✓	✓	✓	✓
D.1.2.1	Review of the Project	✓	✓	✓	✓	✓	✓	✓	✓
D.2.1.1	Implementation Plan	✓	✓	✓	✓	✓	✓	✓	X
D.2.2.1	Documents Prepared	✓	✓	✓	✓	✓	✓	✓	✓

Table 4.6 : Task Documents for Different Purposes

TD ID	Task Document (TD) Name	Model User								Model Re-User		Others Interested	
		The Client	Team/Project Leader	The Modeler	System Supporter	Modeling Supporter	Data Provider	Model Executor	Interested In-house Member	Client/Top-Level Manager	Analysar/Modeler	Researcher/Student	*****
A.1.1.1	Overview of the Client	1,2,3,4,5	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5	1,2,3,4,5	1,2,3,4,5	1,2,3,4,5	1,2,3,4	1,2,3,8	1,2,3,8	1,2,3,8	1,2,3
A.1.1.2	Problem & Need Statement	1,2,3,4,5	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5	1,2,3,4,5	1,2,3,4,5	1,2,3,4,5	1,2,3,4	1,2,3,8	1,2,3,8	1,2,3,8	1,2,3,8
A.1.1.3	Project Team	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4	1,2,3	1,2,3	1,2,3	1,2,3
A.1.2.1	Form of Review	1,2,3,4,5	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5	1,2,3,4,5	1,2,3,4,5	1,2,3,4,5	1,2,3,4	1,2,3	1,2,3	1,2,3	1,2,3
A.1.2.2	Review Outcomes	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5	1,2,3,4,5	1,2,3,4,5	1,2,3,4,5	1,2,3,4	1,2,3	1,2,3	1,2,3	1,2,3
A.2.1.1	Problem Objectives	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4	1,2,3,8	1,2,3,8	1,2,3,8	1,2,3,8
A.2.1.2	Model review Objectives	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4	1,2,3	1,2,3	1,2,3	1,2,3
A.2.1.3	Work Plan Objectives	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4	1,2,3	1,2,3	1,2,3	1,2,3
A.2.1.4	Future Use of the Model	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4	1,2,3,8	1,2,3,8	1,2,3,8	1,2,3,8
A.2.2.1	Experiment Details	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4	1,2,3,8	1,2,3,8	1,2,3,8	1,2,3,8
A.2.2.2	Report Details	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4	1,2,3,8	1,2,3,8	1,2,3,8	1,2,3,8
A.3.1.1	System Description	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4	1,2,3,8	1,2,3,8	1,2,3,8	1,2,3,8
A.3.2.1	Model Physical Element&Details	1,2,3,4,5	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4	1,2,3	1,2,3	1,2,3	1,2,3
A.4.1.1	Simplified Model Physical E&D	1,2,3,4,5	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4	1,2,3	1,2,3	1,2,3	1,2,3
A.4.1.2	Excluded Model Physical E&D	1,2,3,4,5	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4	1,2,3	1,2,3	1,2,3	1,2,3
A.4.2.1	Subsystems	1,2,3,4,5	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4	1,2,3	1,2,3,8	1,2,3,8	1,2,3,8
A.4.2.2	Model Layout	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4	1,2,3,8	1,2,3,8	1,2,3,8	1,2,3,8
A.5.1.1	Data Description	1,2,3,4,5	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4	1,2,3	1,2,3	1,2,3	1,2,3
A.5.1.2	Data Omitted	1,2,3,4,5	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4	1,2,3	1,2,3	1,2,3	1,2,3
A.6.1.1	Validity of Project Specifications	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4	1,2,3	1,2,3	1,2,3	1,2,3
A.6.1.2	Dormant Changes in Project Spec	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4	1,2,3	1,2,3	1,2,3	1,2,3
A.6.2.1	Validity of Model Specifications	1,2,3,4,5	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4	1,2,3	1,2,3,8	1,2,3,8	1,2,3
A.6.2.2	Additional TimeScale&Milestone	1,2,3,4,5	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4	1,2,3	1,2,3	1,2,3	1,2,3
B.1.1.1	Documentation Model	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4	1,2,3,8	1,2,3,8	1,2,3,8	1,2,3,8
B.1.1.2	Model Data	1,2,3,4,5	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4	1,2,3	1,2,3	1,2,3	1,2,3
B.2.1.1	Model Details	1,2,3,4,5	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4	1,2,3	1,2,3,8	1,2,3,8	1,2,3,8
B.2.2.1	Verification Details	1,2,3,4,5	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4	1,2,3	1,2,3,8	1,2,3,8	1,2,3,8
B.3.1.1	Details in Operational Validity	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4	1,2,3,8	1,2,3,8	1,2,3,8	1,2,3,8
B.3.2.1	Details in Parameter Validity	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4	1,2,3,8	1,2,3,8	1,2,3,8	1,2,3,8
C.1.1.1	Overview of Experimentations	1,2,3,4,5	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4	1,2,3	1,2,3	1,2,3	1,2,3
C.1.2.1	WUP and Starting Conditions	1,2,3,4,5	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4	1,2,3	1,2,3	1,2,3	1,2,3
C.1.2.2	Run Length	1,2,3,4,5	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4	1,2,3	1,2,3	1,2,3	1,2,3
C.1.2.3	No. of Replications	1,2,3,4,5	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4	1,2,3	1,2,3	1,2,3	1,2,3
C.2.1.1	Summary of Experiment Run	1,2,3,4,5	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4	1,2,3	1,2,3,8	1,2,3,8	1,2,3,8
C.2.2.1	Summary of Results	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4	1,2,3,8	1,2,3,8	1,2,3,8	1,2,3,8
C.2.2.2	Conclusions & Recommendations	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4	1,2,3,8	1,2,3,8	1,2,3,8	1,2,3,8
D.1.1.1	Review of the Results	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4	1,2,3,8	1,2,3,8	1,2,3,8	1,2,3,8
D.1.2.1	Review of the Project	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4	1,2,3,8	1,2,3,8	1,2,3,8	1,2,3,8
D.2.1.1	Implementation Plan	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4	1,2,3	1,2,3	1,2,3	1,2,3
D.2.2.1	Documents Prepared	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4,5,6,7	1,2,3,4	1,2,3,8	1,2,3,8	1,2,3,8	1,2,3,8

Table 4.7 : Task Documents for Different Audiences

➤ Minimal Documentation

The analysis of documentation process portrays that various types of documents have to be prepared to fulfil the different needs of different audiences. However, the summary produced at the end of the analysis shows that the documents can fall into the following two major categories:

- A document for model users: all task documents are accumulated sequentially to produce this type of document. Such a document may be included the format of progress reports, in which special attention required for individual audience are indicated against the appropriate task document.
- A document for model re-users and other interested parties: minimal relevant task documents to enhance the model-reuse are accumulated to produce this type of document. This may be presented in a similar format of the first document.

Although the analysis is carried out comprehensively, the purposes of documents, which are to be prepared at this stage of the project, are for maintaining the project, protecting contingency losses, and enhancing the model reuse. However, the documents to be prepared may vary from one project to another, depending on their complexity and the current status of the project. Nevertheless, preparation of different documents could easily be handled with task documents, once the relevant audience and the purposes are identified. Therefore, 'completion of documents' task delivers the following task document into MRD database.

D.2.2.1 Documents Prepared: with the aim of presenting current status of the project and the documents prepared. The record consists of the following:

- Current status of the project,
- For whom and for what purpose each document was prepared.

4.8 Summary

This chapter has reviewed and analysed the existing standard project procedures and simulation applications in manufacturing, in order to identify primary tasks for documentation of simulation projects. The primary tasks were then examined in detail in order to identify the details that are to be documented. Identification of details has met the objective of the second research question, in compliance with the proposed approach and the norms for creating task documents. With this comprehensive investigation process, 25 primary tasks and 40 task documents were identified within the simulation project life-cycle to satisfy the diversify requirements of different audiences. Having identified the task documents, the needs of different audience were examined in respect of the documentation purpose as well as task documents, for the purpose of presenting the project details. However, the details descriptions of each task documents are left to discuss in Chapter 05.

5. PRESENTATION OF TASK DOCUMENTS

5.1 Introduction

With the analysis of revised simulation project life cycle proposed, the task documents in each primary task were identified in Chapter 04. However, it has presented only a list of task documents. Therefore, a comprehensive analysis of the contents of each task document (task details) is necessary to:

- be aware the significance of each task document in respect of different documentation purposes and audiences that are benefited from documentation;
- investigate how details are to be recorded (i.e. structure of records) while uniformity and consistency of records for any project are being maintained;
- produce necessary guidelines for recoding and presentations;
- determine any alterations required in previous task documents due to changes in particular task document;
- determine how records are stored (i.e. format of deliverables) in the documentation database to facilitate functional and hierarchical communication; and
- resolve how task documents are integrated to perform documentation in concurrent with the model development process.

The analysis of task details is based on the existing literature in typical project procedures and in applications related to manufacturing. Such an analysis is performed in keeping the objectives of the third research question, but leaving the last two issues related to the documentation database for Chapter 06. Therefore, this chapter discusses other issues in respect to the following viewpoints:

- **Explanation:** Each task document is explored to highlight the contents, considering the iterative nature of the project activities and norms established for creating and managing task documents. Appropriate concepts and practices are discussed briefly.
- **Significance:** The importance of the details is discussed very briefly in respect of the purposes of documentation and potential audiences.
- **Structure and Guidelines:** Contents of each record are examined to draw out any relationships among task details. The structure of records is then decided by maintaining the consistency and uniformity of records, wherever possible. The brief guidelines for recording details are also provided, referencing with the explanation.
- **Feedback to Previous Task Documents:** Attempt is made to establish the feed back loop to task documents in the highest level in the project hierarchy so that such affiliation could be represented in the MRD framework and tool that are discussed in Chapter 06.

5.2 Analysis of Task Documents in Phase 01: Project Specifications

5.2.1 Overview of the Client (A.1.1.1)

The client or the organization, which the study is engaged with, may have come from a different background. It could be a manufacturing or a service organization. Apart from the benefits expected from the problem presented, the client could have short and long-term goals. These goals may have some influence on the study. For example, the client may reuse the model in one stage for expansion of the plant with similar facilities. It could have in-house expertise on the system as well as simulation concepts. For example, if the client has in-house members knowledgeable with simulation concepts, the problem presented may have clear objectives. Such environment will in turn less effort for problem analysis, project team selection, project validation, and possible prediction on the future usage of the model.

Significance: The overview of client presents client's details and the organizational background. They enable the document user to be aware where the project was implemented, to understand the potential use of the model in future, and to realize the effort needs to be made at the beginning of the project to analyse the problem presented.

Structure & Guidelines: The details are presented by means of form approach (Form 5.1) to maintain the uniformity of records, but free-form body text is assigned for descriptive details. The guidelines for details are given within the area that the details are recorded.

CLIENT DETAILS:			
Location :	<i>State name and address of the organization.</i>	Contact:	<i>Name/Telephone/Email</i>
BUSINESS GOALS :	<i>State long/short- term future goals expected in respect of the system under study (but not the goal of the simulation project).</i>		
SYSTEM EXPERTISE :	<i>State availability of in-house expertise in respect of the system or problem of the study.</i>		
SIMULATION EXPERTISE :	<i>State availability of in-house expertise in respect of simulation concepts and practice.</i>		

Form 5.1 : Overview of the Client

Feedback to Previous Task Documents: No previous task deliverables.

5.2.2 Problem and Need Statement (A.1.1.2)

A need of the client may come from either written or verbal description or mixture of them. They may present a brief description about the problem or system together with possible cause and effects and solution techniques with possible illustrations (flow charts, pictures etc.). They may also have some constraints, like, lack of certain resources, capital investment, a time to implement the project etc. However, the client's suggestion on causes

and effects may change as the project progresses, as they have been initially described in terms of the client's background. However, the client always possesses a clear view about the needs of the project. Although, it may find entirely different solutions from that the client brought forward, the client's expectation from the project remains same, perhaps with few additions.

Significance: Problem and need statement delivers a clear description about the problem, any solution techniques and expected benefits communicated at the beginning. They enable the document user to compare the initial problem proposed and benefits expected with the problem diagnosed and the benefits achievable.

Structure & Guidelines: The details are presented by means of form approach (Form 5.2), with free-form body text for descriptive details. Diagrams or any illustration presented by the client to illustrate the problem are recorded separately as appendix or end pages, but referencing the in relevant descriptions.

PROBLEM DESCRIPTION :	<i>State a clear description of the problem presented by the client. Any diagrammatic presentations are recorded separately as appendix or end-pages, but referencing them in the description.</i>
SOLUTION TECHNIQUE(S) :	<i>State any solution technique(s) proposed by the client, in respect of the problem presented.</i>
ANY CONSTRAINTS :	<i>State any constraints such as time, resources, capital etc., if available, for the simulation project.</i>
EXPECTED BENEFITS :	<i>State clearly short-term or long-term benefits expected from project implementation.</i>
References/ Appendix :	<i>State any references made in the problem description or in other areas and where those references could be found.</i>

Form 5.2 : Problem and Need Statement

Feedback to Previous Task Documents: Any changes in this task documents has no effect on the previous task deliverable.

5.2.3 Project Team (A.1.1.3)

Potential members of a project team were discussed in Section 3.4.1.1. However, the choice of team members varies with the complexity of the problem and other constraints such as cost, individual skills, etc. These individual members have different responsibilities and different tasks to be performed in different phases of the project. The client also needs to work closely with the project, by taking correct decisions, making resolutions and providing immense support, whenever necessary. The client may nominate member(s) in different levels for making decisions as well as to provide other miscellaneous assistance needed. Therefore, individual members of the team are required to identify at this stage to carryout

different tasks, taking their experience, attitude, and other qualities into account. Team selections at the very beginning of the project impress the management, enabling them to ensure the smooth functioning of the project.

Significance: According to the purpose of MRD, the description of team members, the reason for individual selection and their responsibilities are significant factors for document user to understand how and in what level team members contribute towards the project.

Structure & Guidelines: The key contents of the record shows direct relationship from one to another detail. Therefore, a specific format is required to show such affiliation. Hence, a form approach with table format (Form 5.3) is suggested to record key contents against each member of the team. The guidelines for details are provided below the form.

TEAM MEMBER	DESCRIPTION	ROLE	REASON (S) FOR SELECTION
The Client/Nominator(s)			
Team/Project Leader			
Model Author(s)			
System Supporter(s)			
Modelling Supporter(s)			
Data Provider(s)			
Model Executor			

Form 5.3 : Team Members

- **Description** (of team members): *Description of individual team members may contain details such as name, present status (designation, authority), etc.*
- **Role** (of team members): *The role of each member may consist of one or more functions like project coordination, system investigation, etc. in broader sense.*
- **Reason(s) for Selection:** *Reason for selection may be due to one or more of the qualities such as depth knowledge and experience in specific areas, authority, attitude, and interests.*

Feedback to Previous Task Documents: Any changes in this task documents has no effect on other previous task deliverables.

5.2.4 Form of Review (A.1.2.1)

It is a fact that the client always holds a clear mind about the output from the project, but the problem presented may naturally have an unclear interpretation. Regardless of the view, it is the responsibility of the project team to analyse and review the problem in-depth with relevant parties, with the aim of finding the real problem, alternative solutions and achievable project benefits and other prevailing constraints. The problem and its background can be reviewed in many ways like interviews, workshops, formal discussions,

brainstorming, referencing etc. However, the approach for reviewing the problem may vary from one project to another, depending on the complexity of the problem as well as the accuracy of the problem presented. The approach is also influenced by the experience of the simulation team. The analysis may begin with brainstorming on the problem presented and then formal discussion with relevant parties - walkthrough approach - to trace specific problem areas that are to be addressed. These specific problem areas may then be analysed through formal review methods with proper schedule and planned queries.

Significance: The records that illustrate how the problem was reviewed and the outcomes from the review help the document user to comprehend the effort to be made for analysing the problem and be aware the direction of the project.

Structure & Guidelines: The details are presented by means of a form (Form 5.4) with free-form body text for descriptive details. However, a need may arise to accumulate those forms, as several reviews may be required to conduct for different issues in a single project.

METHOD OF REVIEW :	<i>State the method applied to review the problem/clear specific issue(s) (ex: an interview).</i>
PERSONALS PARTICIPATED:	<i>State team member(s) and participant(s) attended for the review related to particular issue(s) (ex: a manager in supply department).</i>
ISSUES REVIEWED :	<i>State the specific issue(s) reviewed related to the problem presented. (Ex: how does material delivery affect the through-put time?).</i>
REVIEW SUMMARY :	<i>Summarize the outcomes of the review. (Ex: there is a strong possibility for high throughput time is due to poor material delivery).</i>

Form 5.4 : Form of Review

Feedback to Previous Task Documents: Any changes in this task documents has no effect on other previous task documents.

5.2.5 Review Outcomes (A.1.2.2)

Problem review is carried out, either to clarify the accuracy of the cause and effects presented with the need statement, or to find out possible other causes and alternative solution techniques to the problem. In this process, some of the outcomes from the review may not have a much effect on the problem presented. Some of them may add much complexity to the problem. Some of them may be avoided with simple assumptions. Some solutions may also seek with assumptions. Therefore, it is the task for the team to analyse the outcomes of each review method and present mostly affected outcomes for the problem or system concerned, before the problem objectives are established. The outcomes may generally associate with any alteration to the solution techniques, any assumptions to solution techniques or any changes to expected project benefits.

Significance: If the document presents any alterations made to the problem, cause and effects and solution techniques together with assumptions made and achievable benefits, such details help the user to visualize the deviation of the problem, solutions, and benefits.

Structure & Guidelines: The details are presented by means of form approach, shown in Form 5.5, with free-form body text for descriptive details. However, the review outcome may vary and produce different set of relationships with the contents. The document developer needs to decide how they are presented within the format proposed appropriately.

ALTERATION TO PROBLEM :	<i>State any alterations proposed to the problem presented initially. (Ex: problem is not only in assembly area but also in painting area).</i>
ALTERNATIVE SOLUTION :	<i>State any alternative solutions to the problem, apart from what initially proposed (Ex: change in layout instead of increase in resources).</i>
CONSTRAINTS :	<i>State any constraints discovered during the review that affect the problem or solution techniques.</i>
ASSUMPTIONS :	<i>State any assumption made in respect of problem or solution techniques proposed.</i>
BENEFITS ACHIEVABLE :	<i>All achievable benefits with project implementation. (This may be alteration or addition to the benefits expected in need statement).</i>

Form 5.5 : Review Outcomes

Feedback to Previous Task Documents: Although any change in this task document has considerable effect on the problem and need statement, such changes need not to take into account for illustrating the deviation of the problem and need statement. However, any change in the team members is accommodated at this instance.

5.2.6 Problem Objectives (A.2.1.1)

Problem objectives generally show what output exactly expected or to be achieved at the end of the project. As proposed by Robinson (1994), without any exception, all problem objectives should have clear indication about what is required to achieve at the end of the project and this factor is know as ‘achievement’ of the objective. They may or may not have some kind of ‘measurement’ to measure the achievement, as well as ‘constraints’ to represent the conditions, under which the achievement is made. For example, ‘to reduce the waiting time by 5% without employing more than two operators’ has all components in an objective. Whereas ‘to select the best way of organizing labour that gives the highest output’, has both achievement and measurement; and ‘to demonstrate a new plant’ has only the achievement. For a particular project, there may be more than one objective and, in such situation, the objectives are ranked with their importance with respect to the expected benefits.

Significance: The task document containing clear problem objectives is a vital record for anybody interested with the project. Without knowing those characteristics of the problem objectives, it is hardly believed that any interest for the project stay alive.

Structure & Guidelines: As the key contents of the record may hold relative link among the characteristics, a specific structure is required to show such a relationship. Therefore, form approach with table format (Form 5.6) is suggested to record the details and rank of each objective. However, a need may arise to accumulate number of rows to the table, when more than one objective exists. Brief guidelines for details are presented inside the table.

ACHIEVEMENT	MEASUREMENT	CONSTRAINT	RANK
<i>State clearly the achievement of each problem objective.</i>	<i>State the measurement to indicate the level of the achievement, if available.</i>	<i>State any constraint to point out the condition under which the achievement is mad, if available.</i>	<i>Rank objectives taking the expected benefits into account; if more than one problem objective exists.</i>

Form 5.6 : Problem Objectives

Feedback to Previous Task Documents: Significant deviation from the main aim will have an impact on ‘from of review’ task deliverable.

5.2.7 Model Review Objectives (A.2.1.2)

During the experimentation, a few or a large number of experiments with different number of replications needs to be performed, depending on the nature of the project and the accuracy of the results expected. When the number of experiments and replications increases and the model becomes larger and complex, the run-speed in experimentation will be an important factor to be considered in order to keep the time-scale alive. According to Robinson (1994), nearly 35% of the project time takes for experimentation. Number of experiments and replications can hardly be changed to reduce the time for experimentation. However, run-speed can be considered for any alteration to keep the project time-scale intact, but keeping the accuracy of the model results up to the level of with expectation.

The run-speed is also affected due to the graphical visual display, which could range from none to fully animated 3D display. The management and other parties may so enthusiastic on model demonstration with high graphical contents. But, it is a fact that more the visual display occupies; more the time takes on experimentation and, thus, the project time-scale. Any type of display not only influence for demonstration time, but also for modelling time. For modelling, model building should take place with prior planning to enclose relevant graphical objects. However, if the model is primarily a communication tool, it has no choice left to enclose a less degree of graphic displays. It is also a fact that graphical display provides an aid for understanding the model behaviour, which is a part of model testing

process. Therefore, at this stage, rough decision is to be taken on run-speed and degree of visual displays required, in consultation with the relevant parties and the project time-scale.

Significance: The records that comprise of rough estimate on run-speed and type of visual displays expected are specifically important for management, modellers, and model executors. However, the contained details are subjected to change as the project progresses.

Structure and Guidelines: As the contents in the document are still premature stage to maintain any relationship among the details, a narrative approach is suggested to present details with a choice made by the user. However, in order to provide a checklist, Form 5.7 is proposed to present details, for which guidelines are provided inside the form.

RUN SPEED OF THE MODEL	:	<i>Propose a rough estimate on run-speed, taking experiments, replications, accuracy of results, degree of visual displays and time-scale into account.</i>
VISUAL DISPLAYS	:	<i>Propose a degree of visual interaction required, in consultation with management, model executors, modellers, etc. and in taking the purpose of the project and project time-scale into account.</i>

Form 5.7 : Model Review Objectives

Feedback to Previous Task Documents: No effect on other previous task deliverables.

5.2.8 Work Plan Objectives (A.2.1.3)

In any project, the client likes to hear the status of the project regularly. A most common way of communicating the project status is by conducting progress reviews in different stages, depending on the complexity of the problem. It could be at the end of each major phase or at the end of each critical stage, like problem definition, system investigation, and so on. Project progress may be reviewed with a formal discussion, provided with formal report, which may be an accumulation of selected task documents. Further, a proper work plan is to be established for performing the project effectively, accommodating the time-span that may already have defined on the receipt of the project. This is where the experience of the team plays a key role, which may result, employing additional resources. Similar to the identification of project review stages, the work can be scheduled for each phase or critical stages. However, time scale estimated for review stages and activity plan may not be realistic due to many constraints. Therefore, it may require additional milestones. Regardless of the milestone, the work plan should be included personals involving in each stage. Such identification makes them aware with their role and contribution towards the project.

Significance: The task deliverable may contain records that reside information on major activities of the work and review stages, their milestone with additional ones and personnel responsible for major activities. Such details are vital for document user, particularly for management and team, enabling them to visualize project expectation with time-scale.

Structure & Guidelines: As the key contents of the record has relative link among the details, a specific format is required to show such relationships. Therefore, form approach with table format (Form 5.8) is suggested to record key contents against each other. More details are accumulated by adding rows to the table, in which the guidelines are provided.

MAJOR ACTIVITY (WORK/REVIEW)	PROPOSED TIME- SCALE	ADDITIONAL MILESTONE	PERSANAL INVOLVEMENT
<i>Determine major sequential activities with respect to;</i> <ul style="list-style-type: none"> • <i>Phases or critical work stages, and</i> • <i>Progress review stages.</i> 	<i>Establish time scale for start and/or end of;</i> <ul style="list-style-type: none"> • <i>Work stages, and</i> • <i>Progress review stages.</i> 	<i>Suggest additional milestones for different stages in case of failure to achieve previous.</i>	<i>State person(s) in charge for and other supporting hands for each activity.</i>

Form 5.8 : Work Plan Objectives

Feedback to Previous Task Documents: In establishing realistic time-scale for critical stages, a need may arise to employ additional resources (ex: more data providers) and change in problem objectives (Robinson, 1994). Hence, it may have an impact on ‘project team’ and ‘problem objectives’ task deliverables.

5.2.9 Future Use of the Model (A.2.1.4)

Keller and Dungan (1999) pointed out that it is a normal habit that the models are forgotten once the results of simulation are determined, though they have a second life in the future. Reuse of simulation model for a similar flexible manufacturing system is a good example for such a situation and it can be considered as long-term reuse. Use of a model, which has been developed for designing system facilities, for running day-to-day operational requirements is another example for model reuse. It can be considered as short-term model reuse. Therefore, a potential future usage of the model should be considered at the very beginning of the project to ensure that the model could be reused under different conditions to the system itself or to similar systems with or, without modification.

This is where the client’s short and long-term goals are important in the simulation project. If the management has an intention to use the same model in future for the same system or for a slightly different system, the project team can include some extra effort to make provisions for future expansions or alterations and to ensure that the model can easily be adapted for future usage. One may argue that such a provision should include in project objectives. However, the rule of thumb is that such an inclusion should not affect the current project by any means. Therefore, the future objectives should be listed separately.

Significance: From documentation point of view, the records that contain the potential short-term and long-term future use of the model are important for the document user to understand the model building process, inclusion of additional elements, provision made for future expansion etc. and finally application of the model for some other purposes.

Structure & Guidelines: The details are presented by means of form approach (Form 5.9) with free-form body text for the descriptive details with the choice made by the user.

SHORT-TERM USE :	<i>Brief the potential future use of the model in short-term time frame (ex: use of the model in decision making in day-to-day operation of the system).</i>
LONG-TERM USE :	<i>Brief the potential future use of the model in long-term time frame (ex: use of the model for expanding the system with addition of more resources).</i>

Form 5.9 : Future Use of the Model

Feedback to Previous Task Documents: Any changes in this task documents has no effect on other previous task deliverables.

5.2.10 Experiment Details (A.2.2.1)

Once the model is built to imitate the real-world system, the experiments are performed in changing the values of input variables, which are known as experimental factors. Some experimental factors may simply have binary values like on/off in priority rules, or range of values, like number of machines. These values are referred to as factor range or levels. They can be input into the model in variety ways such as through data files, third party software, simulation model codes and menu driven options, supported by the software. Deciding the methods of data entry is important in planning stage as it would help data collection process and accelerate the model building & experimentation phases. In deciding the experimental factors and their range, some assumptions need to be made to suit the situation. This may have already been addressed and recorded during the review stages. But, repeating such important assumptions here as well highlights their significance. It may also need to perform the experiments through interactive methods depending on the level of visual displays required and the level of experience of the model executor. However, it is assumed that this factor has already been taken into account during the model review objectives.

Significance: The document presents the methods of attaining the problem objectives, how input values are fed into the model and any assumption made to suit the situation. Such details enable the document user to understand the model building process as well as what provision has been made to accommodate different data input methods for experimentation.

OBJECTIVE	EXPERIMENTAL FACTORS	FACTOR LEVELS	METHOD OF DATA ENTRY	ANY ASSUMPTION
<i>Problem objective in brief or identification number.</i>	<i>State experimental factors for attaining the problem objectives set before.</i>	<i>State likely range of values for each factor in form of:</i> <ul style="list-style-type: none"> <i>Max to Min</i> <i>Binary.</i> <i>Sequence, etc.</i> 	<i>Decide the method of data entry with;</i> <ul style="list-style-type: none"> <i>Data files.</i> <i>Third party s/w.</i> <i>Model Code.</i> <i>Menu-driven.</i> 	<i>Any assumptions made in deciding the experimental factors and/or factor levels.</i>

Form 5.10 : Experiment Details

Structure and Guidelines: Table format in the form approach is suggested to record experimental details as the contents of the record has relative link among the details. The suggested format with guidelines is shown in Form 5.10. However, a need may arise to insert additional rows to the table, when the number of experimental factors increases.

Feedback to Previous Task Documents: In deciding experimental details or a change of those details may have in impact on ‘form of review’ and subsequent task deliverables.

5.2.11 Report Details (A.2.2.2)

The goal of the simulation project is to provide the necessary information for the management, enabling them to make decisions. Therefore, at this stage, the project team has to decide what data are to be generated from the experiments. It is true that the main values to be reported are related to the problem objectives. For example, if the objective is to increase the throughput, then the report on throughput becomes the major output. However, failure to achieve the required level of throughput makes the management to look for reasons for failures (e.g. blockage of input buffer). Therefore, it may require additional data to support the potential problem areas, which obstruct the objectives being achieved. Identification of these values to be reported allows the modellers to build the model by making provisions for reporting data in different points of the model. The data can be recorded either with tabular or graphical means to ensure the correct and effective way of communicating results. The effectiveness of the communication also varies with the method of viewing. More visual displays with graphical reports may be useful for the management. Permanent records of data files even with graphical views may be generated using the simulation software itself. These data files may even be imported to third part software.

Significance: The document that contains the values to be reported, the methods of reporting and the method of viewing is important for document user to comprehend the structure of the model and locate where and how the reports are generated from the model.

Structure and Guidelines: A form approach with table format (Form 5.11) is suggested to display a relationship among details. Complete record is a collection of different values to be reported with their details specified in different rows that are added to suggested form.

EXPERIMENT	VALUES TO BE REPORTED	METHOD OF REPORTING	METHOD OF VIEWING
<i>ID for experiment or experimental factor.</i>	<i>State values to be reported in respect of objectives, experiments, and supportive details.</i>	<i>State the method of reporting in the form of;</i> <ul style="list-style-type: none"> <i>• Tabular format,</i> <i>• Graphical format, or</i> <i>• Mixture of them.</i> 	<i>State the method of viewing reports with temporary/ permanent basis, through;</i> <ul style="list-style-type: none"> <i>• Dynamic displays,</i> <i>• Data files, or mixture..</i>

Form 5.11 : Report Details

Feedback to Previous Task Documents: Change in details has no effect on previous tasks.

5.2.12 System Description (A.3.1.1)

At the very first step in the conceptual modelling or system investigation stage, the project team needs to identify the boundaries of the system to be modelled. In this respect, the client may hold an idea with an intension to model everything. However, keeping the focus on problem objectives and future usage of the model and with the knowledge gathered through the problem analysis task; the project team needs to analyse the system proposed by the client in order to identify the exact domain. Therefore, the analysis begins with viewing the overall operations of the system and then omitting unnecessary functional areas from the problem domain. Generally, the choice of boundaries is very closely linked with the goals of the project. However, the problem objectives do not always give an indication about the system or major functional areas to be included or excluded. Therefore, mapping the project goals to the boundaries or the functional areas is based on the team's expertise and common sense. They may simplify the system domain with partial or full inclusion of some functional areas that may have a special influence on the study domain. Similarly, certain functional areas may purposely be omitted due some reasons, or may deals with few assumptions. Those decisions play a key role in understanding the model boundaries and the components.

Significance: The document that contains the details about the major functional areas to be included in the model and special functions to be omitted from the model with potential reasons helps the document user to visualize the domain with approximate boundaries.

OVERALL SYSTEM :	<i>Present the overall picture of the system delivered by the client. Any diagrammatic presentations are recorded separately as appendix or end pages, but referencing in the description.</i>		
MAJOR AREAS SELECTED :	<i>Identify the major functional areas selected.</i>	REASON FOR SELECTION :	<i>State any reason for selection (e.g. objectives)</i>
SPECIAL AREAS OMITTED :	<i>State any special areas that are likely to have affect, but omitted.</i>	REASON FOR OMISSION :	<i>State the reason for omission (e.g. lack of data/future plan etc.).</i>
References/ Appendix :	<i>State any references made in the overall system or major areas and where those references are available.</i>		

Form 5.12 : System Description

Structure and Guidelines: The deliverable of this task is generally a written description, supported by graphical illustrations that can be recorded as appendix. However, the details are presented by means of form approach (Form 5.12) to maintain the uniformity of records.

Feedback to Previous Task Documents: A change in system boundaries may have in impact on 'form of review' and subsequent task deliverables.

5.2.13 Physical Elements and Details to be Modelled (A.3.2.1)

With the details in the system description, the project team may already have overviewed the physical elements of the model. However, from modelling point of view, influences of those elements on the projects objectives are to be evaluated thoroughly, before including them into the model. Some of them may have a certain level of contribution to project objectives. Some may not have any contribution by any means. The elements that contribute to problem objectives may have different contribution levels. Investigation of contribution level of elements as major or minor, not only highlights their importance, but also helps for gradual model development. The details for physical elements vary with the objectives. Deciding the level of detail of elements is a complex issue. In this respect, the project team need to question themselves about the modelling of individual operations of resources, inclusion of set-ups, breakdowns, shift patterns, etc. The simple answer to these questions will be 'yes/no'. However, there is always common reason or special reason behind the decision and some assumptions. This reasoning process may lead to the simplification of model scope and level. However, they are discarded at this stage, leaving to consider them under simplified elements.

Significance: The document that contains a collection of physical elements to be included into the model and checklist of their level of details enables the document user to be aware the scope and the level of the model, before the modelling process begins.

Structure & Guidelines: Form 5.13 is proposed to represent the related details. The details are accumulated by adding relevant rows. The guidelines for details are given below.

TYPE OF PHYSICAL ELEMENT	NAME OF THE PHYSICAL ELEMENT	CONTRIBUTION LEVEL (MAJOR/MINOR)	LEVEL OF DETAILS					ASSUMPTIONS OR ANY SPECIAL REASON FOR MODELLING THE ELEMENT/DETAILS
			INDIVIDUAL OPERATION?	INDIVIDUAL SET-UPS?	SCHEDULE/ SHIFTS?	FAILURES/ BREAKDOWN?	
ENTITY								

PROCESSOR								

STORAGE								

TRANSPORTERS								

CONVEYORS								

Form 5.13 : Physical Elements and Details

- **Type of Physical Element:** *State the general category of the physical element.*
- **Name of the Physical Element:** *Identify a unique name for each type of element so that the same name could be used in the model representation task.*
- **Contribution Level (Major/Minor):** *State the contribution level (major/minor) to represent the importance of the element for modelling.*
- **Individual Operation?:** *Decide whether (yes/no) the operations of individual elements (resources) are to be modelled (not applicable to entities).*
- **Individual Set-ups?:** *Decide whether (yes/no) set-up of individual elements (resources) is to be modelled (not applicable to entities).*
- **Schedule/Shifts?:** *Decide whether (yes/no) the schedule or the shift of individual elements (resources) is to be included (not applicable to entities).*
- **Failures/Breakdowns?:** *Decide whether (yes/no) the failure or breakdown of individual elements (resources) is to be included (not applicable to entities).*
- **(.....)?:** *Addition for new details and then make the decision (not for entities).*
- **Assumptions or any Special Reason for Inclusion the Element/Details:** *State any assumption made or special reason for modelling particular element/details.*

Feedback to Previous Task Documents: In deciding model scope and level or any changes to scope and level may have an impact on the problem objectives and the subsequent task deliverables, particularly for problem objectives.

5.2.14 Simplified Elements and Details (A.4.1.1)

Simplified elements and/or details are accomplished by grouping physical elements into one element, or substituting an element and details for another in a complex situation. Black box modelling is the most common method for grouping resources. For example, a group of machines is represented as a time-delay, which embodies a sum of processing time, in a black-box element without details of individual machines. Such an approach is more suitable when the model encounters many different systems, lack of individual data or insignificant detailed description situation. However, the simulation team should sacrifice some output data, like unitisation of individual machines, queue statistics, etc. Moreover, in a high volume manufacture, a group of entities can be considered as a single entity to improve the run-speed of the model. However, again, it restricts assigning of different attributes to individual items and, therefore, it loses the modelling flexibility and output. According to Robinson (1994), modelling of deliveries, instead of transportation, is a good example for substituting. Author pointed out that when less detail for transporters are available, it can be substituted by a delivery element comprised of details, like a number deliveries, average movement time etc. Someone can argue that substitution process is a

form of black box modelling. However, in the above example, characteristics such as traffic congestion, drivers' shifts, breakdown etc. that are expected in a transportation system are all replaced by an element with relevant simplified data. Therefore, both grouping and substitution of physical elements are considered at this stage simultaneously for the purpose of documentation.

Significance: If the document provides the information on the simplified elements/details with representing elements/details, reasons for simplification, any assumption made regarding the simplification and the results that are likely to affect, then such details enable the document user to be aware how the individual elements are represented in the model.

Structure & Guidelines: Form approach with table format (Form 5.14) is proposed to deliver related details. The details are accumulated by adding rows.

E&D TO BE GROUPED/ SUSTITUTED	REPRESENTING ELE & DETAILS	REASON S AND ASSUMPTIONS	ANY RESULTS BE AFFECTED
<i>State the name of element(s) (from Form 5.13) to be grouped or substituted and brief details to be replaced.</i>	<i>State the name of the element to represent the element(s) groped or substituted and brief representing details.</i>	<i>State the reason for grouping or substitution and any assumptions made.</i>	<i>State only major results be affected due to simplifications.</i>

Form 5.14 : Simplified Element and Details (E&D)

Feedback to Previous Task Documents: Simplified model elements and details may have an impact on the model review objectives and the subsequent task deliverables.

5.2.15 Excluded Elements and Details (A.4.1.2)

Model elements and details are sometimes needed to ignore when a particular situation that arises or occurs infrequently or in order of pattern, and when good data are unavailable. For example, if a machine in the system is used very rarely during the process cycle or a breakdown of a machine occurs in long time interval, those types of elements and the details can be omitted from the model, as they are infrequent events. Availability of common shift pattern for resources with same capacity is a good example for regular pattern of details. Such details can also be excluded, as they do not contribute much for validity of the model. Regardless of its contribution level to the model, lack of good data often occurs in simulation. This is very common with labour resources. Although, it is too early to make decision on this aspect, a possible prediction of such elements is feasible for model specification. Needless to mention that the team should bear details loss due to such exclusion, with some assumptions or allowances.

Significance: The document that contains the elements and details to be excluded from the model and the reasons for such exclusion with potential output losses enables the document user to comprehend the model elements and details and, thus, the structure of the model.

Structure & Guidelines: Form approach with table format (Form 5.15) is proposed to deliver related details. The details are accumulated by adding rows.

ELEMENTS & DETAILS TO BE EXCLUDED	REASON S AND ASSUMPTIONS	ANY RESULTS BE AFFECTED
<i>State the name of element(s) (in Form 5.13) and details to be excluded.</i>	<i>State the reason for exclusion and any assumptions made.</i>	<i>State only the major results be affected due to exclusion.</i>

Form 5.15 : Excluded Element and Details

Feedback to Previous Task Documents: The elements and details excluded from the model may have an impact on the model review objective and subsequent task deliverables.

5.2.16 Subsystems (A.4.2.1)

If the system being studied is complex, the easy way of modelling the system is splitting the overall system into subsystems. The subsystems are identified with one or combination of flow, functional or state-change approach (Sadoun, 2000). The flow approach is based on the information flow of the system. If the system were an assembly plant, for example, the flow approach would be more feasible for splitting the system. The second, functional technique is used when there is no clear flow of entities through the system under study. In such a situation, logical sequences of functions being performed are identified and the system characteristics that perform a given function are grouped to form a subsystem. This approach can be used for manufacturing facilities that are arranged functionally. The third, state-change approach is useful for systems that are characterised by a large number of interdependent relationships that need to be investigated regularly to find the state changes.

However, it may always not possible to split the entire system into several subsystems. In such a case, the system may be represented as a collection of subsystems and individual elements. Individual subsystems identified may further be subdivided into lower level subsystems. The significant factor in splitting subsystem is that linking of subsystems into the overall system. The general approach is that subsystems are represented as black-boxes in a model by linking overall data from one sub-model to another sub-model or element, depending on how the system are broken-down. In this model splitting task, the model author takes many decisions and makes assumptions in isolating subsystems as well as about their integration with other subsystems and/or the overall model (Kiran et al., 2001). Those details in subsystems and any assumptions made are important for understanding the model structure and for building the model.

Significance: The document containing the details of subsystems, how they are linked and any assumptions associated in isolation process, enables the document user to builds and interpret the model in the same way that the model was expected.

Structure & Guidelines: Form approach with table format (Form 5.16) is proposed to deliver linked details. The details are accumulated by adding rows.

NAME OF SUBSYSTEM	ELEMENTS & NEXT LEVEL SUBSYSTEMS	INPUTS TO SUBSYSTEM	OUTPUTS FROM SUBSYSTEM	TECHNIQUES AND ASSUMPTIONS
<i>State name for the subsystem isolated.</i>	<i>State the next low-level subsystems and physical elements, if any, to be included into subsystem.</i>	<i>State the inputs to the subsystem.</i>	<i>Identify the output expected from the subsystem.</i>	<i>Briefly present the techniques used in isolation and any assumptions made.</i>

Form 5.16 : Subsystems

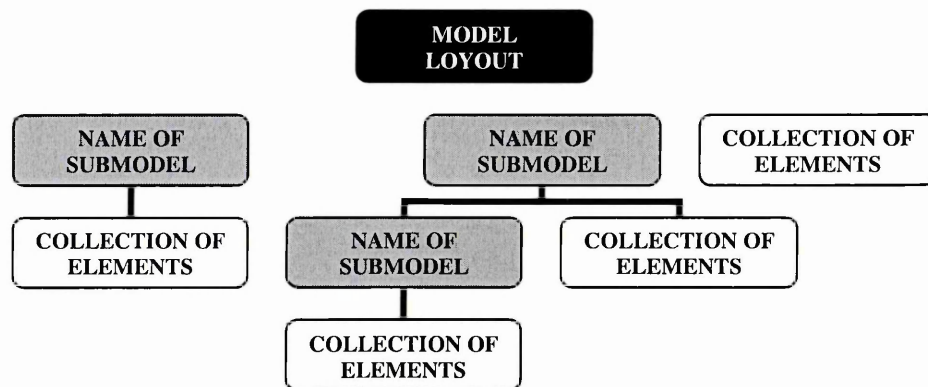
Feedback to Previous Task Documents: Change in details has no effect on previous tasks.

5.2.17 Model Layout (A.4.2.2)

The subsystems task document shows the background details of isolation process of subsystems. However, their relationship with overall model is essential to present the overall picture of the model structure with respect to physical elements. Such a representation also leads to an incremental development of the model by either vertical-horizontal or horizontal-vertical approaches (Randell, 1999). The hierarchical diagram is the best solution for this purpose, as it represents the increased level of details in downward direction. However, unlike with IDEF diagrams, hierarchical diagram does not represent the interaction between the sub models and the overall model. However, at this stage of the project, such a minimal representation is sufficient for the purpose of the documentation.

Significance: If the document shows the relationships of subsystems (sub-models) and physical elements with the overall model, it helps the document user to understand the model layout and develop the model independently and incrementally.

Structure & Guidelines: A diagrammatic representation [Form 5.17 (a) – sample for demonstration purpose], which is supported by organization chart tool, is used to represent the hierarchical structure of the overall model.



Form 5.17 (a) : Model Layout

The form proposed above could be altered to create different models. To differentiate between sub-models and the individual physical elements, a shaded and non-shaded representation is proposed respectively. However, Form 5.17(b) is proposed to record comprehensive details of the model layout, including individual element names so that the same details could be used for the model representation task through data exchange.

MODEL LEVEL NAME	MODEL LEVEL NUMBER	NAMES OF SUB-MODELS	NAMES OF PROCES-SORS	NAMES OF STORAGES	NAMES OF TRANSPOR-TERS	NAMES OF CONVEY-ORS
<i>Model or Sub-Model Name.</i>	<i>Model/Sub level No. in the hierarchy (e.g. 1.1).</i>	<i>Collection sub-models names, separated by comma.</i>	<i>Collection of processors names, separated by comma.</i>	<i>Collection of storages names, separated by comma.</i>	<i>Collection of transporters names, separated by comma.</i>	<i>Collection of conveyors names, separated by comma.</i>

Form 5.17 (b) : Model Layout Details

Feedback to Previous Task Documents: Change in details has no effect on previous tasks.

5.2.18 Data Description (A.5.1.1)

This task document generally should provide the information for data providers on what data are to be collected, what data are to be estimated, sources from which they are collected, how they are estimated, any alternative sources for data collection or methods for data estimation, who is responsible for data collection and when they are to be readily available. Providing information to data providers are only one aspect of this document. It should also provide the other team members about the accuracy, reliability, and integrity of the data. It is also important to identify that how data are input into the model so that total population could be represented and the data provider could follow the data format expected. Such formatting was encouraged by simulation software that provides flexibility to input data through menu driven options, direct data files as well as third party software. If data cannot be transformed into correct format, they are treated as inaccurate data and simulation results needs to be judged accordingly (Robinson, 1994). In this complex process, the project team make many decision and assumptions to suit the present situations.

Significance: If a document delivers technical description of data, such as the data required, the data sources, the purpose of data and assumptions, as well as management details, such as individual responsibilities and time scales, such a document is important for communication, reporting and future usage to envisage the data acquisition process.

Structure & Guidelines: Form 5.18 with table format is proposed to record related details. The details are accumulated by adding relevant rows. The guidelines for completing the form are provided below the table. The generic details shown in the table (1st and 3rd columns) are only for illustration purpose. The proposed format may be useful to share the details with the model representation task.

TYPE OF ELEMENT	ELEMENT NAME	TYPE OF DATA ITEM	PURPOSE*	INPUT FORMAT*	SOURCE/HOW ESTIMATED	ALTERNATIVE SOURCES/WAYS	APPR % ACCURACY	ANY ASSUMPTIONS	PE. RESPONSIBLE*	FUTURE CHANGE EXPECTS?
PROCESSOR		SCHEDULE								
ARRIVAL		TIME BET ARRIVALS								
RANKING		CRITERIA								
-----		-----								
* KEYS	PURPOSE		INPUT FORMAT		PERSONAL RESSPONSIBLE					
	1.	For model building.	1.	Data file.	1.	(Define)				
	2.	For experimentation.	2.	Menu driven.	2.	(Define)				
	3.	For validation.	3.	Model code.	3.	(Define)				
	4.	(Define)	4.	(Define)	4.	(Define)				

Form 5.18 : Data Description

- **Type of Element:** State the generic type of the element (Tables 4.1, 4.2, 4.3 & 4.4).
- **Name of Element:** State unique name for element (e.g. grinder in processor element).
- **Type of Data Item:** State the generic type for the data item as in Tables 4.1, 4.2, 4.3 and 4.4 (e.g. schedule in grinder processor element).
- **Purpose:** State the purpose (model building/validation/experimentation/other) of the particular data item. This will give an indication for time scale.
- **Input Format:** Decide how data are input into the model (menu driven, model code, data file, etc.)
- **Approximate % Accuracy (expected):** Approximate accuracy of data, based on the project team judgement.
- **Sources/How Estimated:** State the source(s) of data to be collected. If it fails to collect, state how they are estimated.
- **Alternative Sources/Ways:** Identify any alternative sources for data collection or alternative ways for data estimation.
- **Any Assumptions:** State any assumptions made any other special reasons in respect of data acquisition, sources, accuracy etc.
- **Personal Responsible:** State who is responsible for data acquisition process.
- **Future Change Expects?:** State any future expectation of data changes that have an impact on model results. If yes, state what impact and how such changes are handled.
- **KEYS:** Keys (refer the table) are used to present information conveniently.

Feedback to Previous Task Documents: There is no effect on previous task deliverables due to changes in details of the record.

5.2.19 Data Omitted (A.5.1.2)

When data is unobtainable due to non-availability of sources or failure to estimate accurate data, those data should be omitted or could be replaced by some other type of data. Replacement of variation of cycle time by a fixed cycle time is good example for the last option. The last option may have already been dealt during model scope and level reduction tasks and recorded in both 'simplified element and details' and 'data description' task documents. Therefore, the significant in this task document is to record the data omitted from the model, though it is early to predict such a situation. Even though, this is the worst-case scenario, according to Robinson (1994), it often happens in simulation projects. Therefore, a careful consideration on this aspect, before the model building commences, is not a waste of time, as the elimination of data may lead to further simplification of model scope and level or may even change the problem objectives. Without a doubt, data omission leads to the loss of accuracy of the model results. However, there should be valid reasons for such a data omission and it may have even been supported with assumptions.

Significance: The document, which contains the data items excluded, the reasons for such omission, any assumptions made, and potential results that are likely to affect, enables the document user to comprehend accuracy of the model and the results.

Structure & Guidelines: Form approach with table format (Form 5.19) is proposed to deliver related details. The details are accumulated by adding rows.

ELEMENT TYPE/NAME	DATA ITEMS OMITTED	REASON/ANY ASSUMPTION	ANY RESULTS BE AFFECTED
<i>State the element name (ID).</i>	<i>State the generic name of data item omitted.</i>	<i>State the reason for omission or any assumption made.</i>	<i>State results that are likely to affect due to omission of data item.</i>

Form 5.19 : Data Omitted

Feedback to Previous Task Documents: Any omission of data may lead to changes in the problem objectives and the subsequent task documents.

5.2.20 Validity of Project Specifications (A.6.1.1)

Activities in a simulation project are generally iterative by nature. When a certain task completes, a need may arise to alter the details of previous task documents. Although, many progress review stages are conducted previously with the management, project specifications are still need to be validated. Details to be validated with management generally include the project objectives (problem, progress review, work plan, future use); the benefits achievable, experiment and reports details and perhaps the elements and data simplified or omitted as they may have useful contribution to the final results. Those details have already been recorded and updated during sequential as well as iterative tasks. The

specifications that are to be presented for validation vary with the participants attended in validation process as well as their contribution level towards the project. For example, if the participants have an in-depth understanding of system details, they may also show their vested interest on model details. Therefore, the project team need to decide appropriate details that are to be presented for validation. Regardless of the review stages conducted previously, it is not strange, if the project specifications are changed in this stage. But, they are to be recorded and subsequently previous task documents are to be altered accordingly.

Significance: If a document delivers the project specifications validated, the participants in validation and any alteration proposed to the specifications, such a document is important for communication, future references in model implementation stage and future use of the model.

Structure & Guidelines: The details are presented by means of form approach (Form 5.20) with free-form body text for descriptive details with a choice by the document developer.

PROJECT SPEC VALIDATED :	<i>Provide a list (a collection of task documents) of specifications presented for the validation process.</i>
PARTICIPANTS FOR VALIDATION :	<i>State the participants (members of the team, in house members, outside participants, experts etc.) for validation process.</i>
PROPOSED ALTERATIONS TO PROJECT SPEC :	<i>State any alteration proposed (whether they can be handled or not) for the details presented for validation process.</i>

Form 5.20 : Validity of Project Specifications

Feedback to Previous Task Documents: Any changes in specification may have an impact on problem and need statement and subsequent task deliverables.

5.2.21 Dormant Changes in Project Specifications (A.6.1.2)

Project validation process may suggest alterations to the project specifications. According to Robinson (1994), such changes may arise due to changes in real-world, increased understanding of the potential benefits in the simulation project or identification of new problem as the project progresses. However, occupying such changes may increase the project time and cost. Therefore, the management should accept or reject the proposed changes, considering the factors like their importance, available constrains, loss of accuracy of results, etc. If the proposed alterations are accepted, then the project specifications are re-issued and comments are re-sought. If they are not accepted due to certain reasons or are decided to handle in some other way (e.g. delaying them for the next project or the future development, ignoring with assumptions, etc.), they need to be recorded for future references. Also, potential pitfalls or consequences, which may cause due to changes suppressed or delayed, are to be recorded as they may have an influence on the final results and the recommendations.

Significance: This document delivers the proposed changes to specifications (but they are handled in some other way), how such changes are handled, and potential effect on the model results due to suppression or delaying the suggested changes. Such records provide the document user with some evidence on the results expected and the future work.

Structure & Guidelines: Form approach with table format (Form 5.21) is proposed to present linked details. The details are accumulated by adding rows.

CHANGES PROPOSED TO SPECIFICATIONS	METHOD OF HANDLING CHANGES	ANY RESULTS BE AFFECTED
<i>State the proposed changes that are to be handled in some other way due to certain constraints).</i>	<i>State how such changes are handled (e.g. ignore them with assumptions, delay, suppress, etc.).</i>	<i>State potential results that is likely to affect, if available, due to the decision made previously.</i>

Form 5.21 : Dormant Changes in Project Specifications

Feedback to Previous Task Documents: There is no effect on previous task deliverables due to changes in the details of this task document.

5.2.22 Validity of Model Specifications (A.6.2.1)

It is essential that the technical details (model specifications) are to be validated with the relevant parties, before proceeding to model building process. Details to be validated in respect of the model generally include the system boundaries; brief description of model scope and level, the model lay out, major assumptions made in respect of model simplification and data collection etc. They could be collected directly from previous task documents. Some details presented in project summary like objectives, experiments, reports etc. could also be repeated for better understanding of the modelling approach. Similar to the project summarization task, the model specifications to be presented for validation vary with the participants in validation process as well as their contribution level. Regardless of the number of reviews conducted previously, the model details may still need to be changed during this validation process. If such changes are proposed, they should be recorded and the previous task deliverables are to be altered accordingly.

MODEL SPEC VALIDATED :	<i>Provide the list (a collection of task documents) of model technical specifications presented for the validation process.</i>
PARTICIPANTS FOR VALIDATION :	<i>State the participants (members of the team, in house members, outside participants, experts etc.) for validation process.</i>
PROPOSED ALTERATION TO MODEL SPEC :	<i>State any alteration suggested for model specifications presented for the validation process.</i>

Form 5.22 : Validity of Model Specifications

Significance: The document that comprised of the model specifications validated, the participants for model validation process, and any alteration made to the model specifications, helps the document user to aware the credibility of the model to be built, before hand.

Structure & Guidelines: The details are presented by means of form approach (Form 5.22) with free-form body text for descriptive details.

Feedback to Previous Task Documents: Any changes in specification may have an impact on the model physical elements and details and the subsequent task deliverables.

5.2.23 Additional Time-scale and Milestone (A.6.2.2)

Once the model specifications are agreed, additional time-scale and milestone could be established to keep the track on the project progress. The time-scale for major activities such as model building process, experimentation etc. as well as progress review stages has already been set up during the objective identification task. In addition to major time-scale that have already been set up, micro-level time-scale could be established for sub models so that the team (particularly modellers) could be aware the benchmarks in the incremental model development process. This may be particularly useful for a large-scale project.

Significance: If the document delivers time-scale established and individual responsible for different sub models, such records provide the information for the management as well as the team to track the project status any given time.

Structure & Guidelines: Form approach with table format (Form 5.23) is proposed to deliver linked details. The details are accumulated by adding rows.

NAME OF SUB-MODEL	RESPONSIBLE PERSONAL	TIME SCALE
<i>State the name of the sub-model.</i>	<i>State who responsible for model development.</i>	<i>State the target time-scale established.</i>

Form 5.23 : Additional Time-scale & Milestone

Feedback to Previous Task Documents: There is no effect on previous task deliverables

5.3 Analysis of Task Documents in Phase 02: Model Building and Testing

5.3.1 Documentation Model (B.1.1.1)

Documentation model is a graphical representation of the executable model and is constructed with predefined generic reusable model elements. If the technique adopted to represent the model is neutral technique such as flowcharts, IDEF, ACD, UML etc, then the documentation model may be considered as generic and software-independent representation of the model. Such representation makes a provision to translate documentation model into software-dependent model. Nevertheless, the use of generic model elements and neutral techniques with its stipulated symbols may reflect a conflict in usage. Thus, amalgamation of symbols in neutral technique and model elements is required.

- **Flowcharts and Model Elements for Documentation Model (DM)**

The benefits of model representation depend on the techniques or tools used for it. As reviewed in Chapter 02, the flowcharts technique offers many advantages compare with other neutral techniques in respect of model documentation. However, it has many pitfalls as described in Section 2.5.2.5. Nevertheless, the ability to customize the flowcharts to fit into particular application creates a flexible environment to integrate flowcharts and its features with reusable model elements. Hence, it appears that flowchart is the most appropriate choice for the construction of DM. Therefore, the reusable model elements are linked with flowcharts basic symbols so that a user-friendly environment could be established for the construction of DM and DM could be translated into the executable model.

In flowcharts, the basic symbols are supplemented by user-defined text for producing a meaningful description. Similarly, the symbols in DM are to be supported by user defined text. However, a common structure for the text description is proposed, not only to maintain the consistency and uniformity of model description, but also to use the same information for model translation. Accordingly, the corresponding flowcharts symbols for different reusable model elements, together with their text description, have been proposed as shown in Table 5.1. The table depicts only a few model elements for demonstration. Also, the data elements are not included in the table, as they present a common layout in the text description that includes the details regarding the type of data items, where they are stored and in what model level.

- **Standard Notation for Model Elements and Terminology**

When the formalised or simplified elements are identified for the DM and the user options are established for rules like FIFO in ranking, cyclical rule in resource selection, etc; common notations are to be employed to maintain the uniformity of documentation model and to facilitate for the model exchange process. However, structural and terminology differences in the existing software raise a need for standard model elements that comprised of standard terms. An attempt has been made to generalize the elements with their details as discussed previously. However, the experience with Arena application and a little exposure with other simulation software may have had some influence on the proposed elements and user options.

- **Construction of Documentation Model**

Construction of the documentation model with the use of symbols defined previously follows a similar approach to model development process, in which the complexities of the model are added step-by-step to ensure the right and accurate model.

Type of Element	Flowchart Symbol for Model Element	Information to be Displayed with Default Value	Proposed Symbol & Structure of Display
Physical Element	New symbol for 'Entity'	<ul style="list-style-type: none"> Unique name for the entity - [Entity Name] 	
Physical Element	New symbol for 'Output'	<ul style="list-style-type: none"> Unique name for the departing item - [Output Name] 	
Physical Element	Customized Process symbol for 'Processor'	<ul style="list-style-type: none"> Unique name for the processor - [Processor] Number processor units - (1) Station ID for which processor attached - {Sta_[Processor]} Process name(s) that processor engages - [Process(es)] Resource cycle ID - {RCy-[Processor]} 	
Physical Element	Customized Process symbol for 'Phy Sub Model'	<ul style="list-style-type: none"> Unique name of physical sub-model - [Phy Sub Model] Level of the sub-model - [1.????] Associated information sub-model name-[Info Sub Model] 	
Primary Information Element	New symbol for 'Arrival'	<ul style="list-style-type: none"> Unique name for arrival - [Arrival] Name of arrival entity - [Entity] Number entities per arrival - [#] Time between arrivals - TBA Time units - [Unit] 	
Primary Information Element	New symbol for 'Depart'	<ul style="list-style-type: none"> Unique name for departure - [Depart] Name of departing item - [Output Name] 	
Primary Information Element	Customized Process symbol for 'Process'	<ul style="list-style-type: none"> Unique name for the process - [Process] Name of the set or processor(s) attached to process - {a Set/Processor(s)...} Process time ID as Attribute - {A-Time_[Process]} 	
Primary Information Element	Customized Process symbol for 'Info Sub Model'	<ul style="list-style-type: none"> Unique name of information sub-model - [Info Sub Model] Level of the sub-model - [1.????] Associated physical sub-model name-[Phy Sub Model] 	
Auxiliary Information Element	Customized Preparation symbol for 'Set'	<ul style="list-style-type: none"> Type of the set (Processor/Station...) - [Processor] Unique name for the set - [Set] Members of the set in order - [Members] 	
Control Element	Customized Process symbol for 'Ranking'	<ul style="list-style-type: none"> Operation/Capture name associated with raking - [Ope/Cap] Ranking rule - [Rule] Attribute/Variable name associated with rule - [A-Att/V-Var] Tie breaker, if rule failed - [TieBreaker] 	
Control Element	Customized Process symbol for 'Selection'	<ul style="list-style-type: none"> Set name associated with selection - [Set] Operation/Capture name associated with selection - [Ope/Cap] Set member selection rule - [Rule] Attribute/Variable name associated with rule - [A-Att/V-Var] 	
Control Element	Customized Decision symbol for 'Decision'	<ul style="list-style-type: none"> Attribute/Variable or Initial expression name that decision based-on with specific format - [A-Att/V-Var/E-Exp] Type of evaluator - [Eva] ID for attribute/variable/expression value - [ExVal] 	
Control Element	Customized Multi-In/Out for 'Distribution-In/Out'	<ul style="list-style-type: none"> Percentages/Quantities In/Out - [Values] 	
Control Element	Customized Parallel Mode symbol for 'Priority'	<ul style="list-style-type: none"> Unique name for priority - [Priority Name] Operation and/or Capture names with priority values - [Ope/Cap Name-1/2/3,...] 	
Logical Element	Customized Process symbol for 'Capture'	<ul style="list-style-type: none"> Unique name for capture - [Capture Name] Set or Resource(s) to be captured - [Set/Resource(s)] 	
Logical Element	Customized Process symbol for 'Free'	<ul style="list-style-type: none"> Unique name for free - [Free Name] Set or Resource(s) to be freed - [Set/Resource(s)] 	
Logical Element	Customized Process symbol for 'Assign'	<ul style="list-style-type: none"> Assigned data type (e.g. process times) - [Assign Data Types] Entity name or other stages of entity - [Entity (Stage) Name] 	

Table 5.1 : Symbols of Documentation Model Elements

Modellers naturally tend to begin the model development with physical elements, based on the systems physical layout and schematics diagrams that are already available. Following this natural approach, it is proposed that the physical elements are represented at the beginning of the construction process of DM. This lays a foundation for the document developer to construct the process flow of the system with the information elements. Primary information elements are then apprehended into the DM while the physical elements are being coupled with them to establish relationships. In establishing the relative links, a need for auxiliary elements may arise and they are then added in succession.

Once the process flow is constructed, the control elements are added to the model whilst meaningful identities or expressions are being added for better understanding of the model behaviour. However, accumulation of both information and control elements simultaneously accelerates the construction process. Once the model behaviour is captured by physical, information, and control elements, logic elements are added to represent model logics and to make relationship with data elements. Finally, data elements are added to the top level or to each level of the model to produce formatted data tables with predefined and supplementary details so that the user could complete them appropriately. Although, the steps in this process are presented sequentially, it is natural that representation of composite object makes the construction process more efficient.

The symbolic representation, which is compatible with basic processes in the construction of flowcharts, provides a more convenient environment for the document developer to record details, based on the modeller's natural thinking. The information collected in this way is useful not only for model documentation, but also model translation.

Significance: The documentation model presented in any form enables the document user to comprehend the model and model logic explicitly

Structure & Guidelines: A generic structure for any type of model cannot be proposed. However, the use of graphical symbols and guidelines discussed previously produce consistent and uniform documentation model. It can be recorded as graphical object inside the Form 5.24, provided with a user description to differentiate different level of the system.

Insert the documentation model as a graphical object. If the model has more than one level, individual sub-models can be named appropriately within this form.

Form 5.24 : Documentation Model

Feedback to Previous Task Documents: Any alterations to the model elements and their level of details may raise a need for changing the 'elements and details to be modelled' and the subsequent task documents.

5.3.2 Model Data (B.1.2.1)

Process times, resource schedules and failures, transport distances etc. are generally considered as model data. They can be directly recorded for the purpose of documentation. For example, process times of a particular entity can be recorded by using the visiting processors, data values, data units, and, perhaps, with the sequence. However, for documentation, the data recorded should benefit to other tasks, particularly to the model coding process. In other words, the documentation model should be able to translate into the executable model. Some data items can be used to drive the model. For example, the process times can be used as attributes for the construction of model logic. Considering these characteristics, process times are recorded in a predefined format so that those data tables could be interfaced with the relevant model elements to drive the model. In this respect, the data elements of the documentation model, which produce a specific format for recording the data, interacting with other elements, play a vital role in the integrated documentation system. Other data items can be presented in the similar way to make the documentation process integrated and productive. For illustration purpose, the format proposed for recording process time data is shown in Table 5.2.

MODEL LEVEL	ENTITY/ENTITY STAGE NAME	ASSOCIATED ELEMENT TYPE	ASSOCIATED ELEMENT NAME	EXPRESSION OR ID	EXPRESSION TYPE	DATA VALUES	DATA UNITS
<i>State the model level name (e.g. subModel1)</i>	<i>State the element name (e.g. Part A) or entity stage name (Rework Part A) to link with the model</i>	<i>State the generic name of the element type (e.g. process) to keep the consistency with other records</i>	<i>State the element name associated with process time (e.g. grinding)</i>	<i>State ID or expression for process time. The format proposed is Time_ElementName.</i>	<i>State how the process time is treated in the model (always it is an Attribute).</i>	<i>State data values in a standard format.</i>	<i>State data units in a standard format.</i>

Table 5.2 : Process Times

In the above table, columns except the last two (data values and units) are extracted using the 'process times' data element in the MRD tool developed. Also, a standard format for last two columns is proposed so that the document developer could update them with appropriate data values and units. Similar format with slight variation can be used for other types of model data and they are illustrated in the case study (Annexure C).

Although the above description deals with direct data items, from documentation point of view, other assign attributes like priorities in queue controller; variables like waiting time in queues; and any expressions that are associated with attributes or variables can also be considered as model data. For instance, the expressions incorporated in control elements as attributes, variables or expressions are recorded with a specific format (see Table 5.3) to facilitate to model exchange process.

MODEL LEVEL	ASSOCIATED ELEMENT TYPE	ASSOCIATED ELEMENT NAME	EXPRESSION NAME	EXPRESSION VALUE	DESCRIPTION
<i>State the model level name (e.g. subModel1)</i>	<i>State the generic name of the element type (e.g. decision) to keep the consistency with other records</i>	<i>State the element name associated with expression (e.g. Queue Selection)</i>	<i>State ID or expression name used in particular element for capturing model logics</i>	<i>State the value in respect of the software dependent executable model or keep it blank</i>	<i>Explain the meaning of the expression and the how & where the values presented in the executable model</i>

Table 5.3 : Expressions

Similar to process time, all details except the last two columns are derived from the documentation model with the use of 'expression' data element. The expression values in fifth column may correspond to the particular software being used. Hence, they are to be updated after the model development.

Similar to the need identified about the standard notations for model elements and user options, in presentation of data values and units, the question 'what are the notation that the user is allowed to use as input data values and units?' will rise. Although, notations proposed are based on Arena, establishing a standard notation is still an issue to be attended.

Significance: The document, which contains model generic data, indicating how they correspond with the model and other assigned parameters entrenched with the model, enables the document user not only to be aware about the data associated with model, but also to understand the role of individual parameters in the model.

Structure & Guidelines: A specific format for individual data items can be suggested. However, a common layout cannot be proposed due to different types of details associated with simulation projects. Nevertheless, the data table produced with the use of the documentation model can be imported with appropriate sequence with the choice made by the user, perhaps into the Form 5.25.

Insert data table and name them appropriately inside this form.

Form 5.25 : Model Data

Feedback to Previous Task Documents: Any alterations to the data items (not values) will raise a need for changing 'data descriptions' and the subsequent task documents.

5.3.3 Model Details (B.2.1.1)

The details of the executable model exclusively depend on the software being used and how the modeller attempt to present the model self-explanatory with the use of facilities provided in the software and other guidelines. Also, the records, such as input data and text or graphical output associated with data and programmes (model codes) delivered from the software may vary with the software used. Therefore, it is much more difficult to propose a common layout for recording the model details. However, from documentation point of view, records of the details associated with the executable model in any form are still valuable. Furthermore, recording file names of the model and other related information is useful for future references and traceability.

Significance: If the task document delivers what software was used for the model and the other supporting details, where the model and associated details were stored, and the other potential details either in graphical or text format; such records help the document user to trace the information, visualize how the model was built, identify other details (e.g. data, collection of elements, etc.) associated with the executable model.

Structure & Guidelines: Form 5.26 is suggested to maintain the consistency of records, but with free-body text format. A need may arise to present the comprehensive model details with few diagrams or tables like graphical model, element list, etc. depending on the facilities provided by the software used. In such a case, the information is recorded separately as an appendix or end pages, but referencing them in appropriate descriptions.

SIMULATION SOFTWARE :	<i>Name of the software used.</i>	MODEL FILE NAME :	<i>Filename with extension.</i>
OTHER SOFTWARE (If, available) :	<i>Name(s) of any other software associated in model building.</i>	SUPPLEMENTARY DATA FILENAME (If, available) :	<i>Filename(s) associated in supplementary details.</i>
MODEL CODES :	<i>State any description about the model codes to support the overall picture of the model and where the hard copies of model codes (graphical or text or both) are available in final documents.</i>		
ANY SUPPLEMENTARY DETAILS :	<i>State any details extracted from the model to support model details. Any diagrammatic or text presentations are recorded as appendix or end pages, but referencing in the description.</i>		
References/ Appendix :	<i>State any references made in the model codes and/or in supplementary details and where those references are available.</i>		

Form 5.26 : Model Details

Feedback to Previous Task Documents: Any addition or subtraction of model elements, beyond the model scope and level established before may result to change the problem objectives and the subsequent task deliverables.

5.3.4 Verification Details (B.2.2.1)

As the verification associates with micro-level of checking (white-box validation), the primary concern at this stage is the validity of individual elements and their data to ensure that they behave in the manner intended by the model code. For this purpose, there are many static and dynamic techniques can be found in the literature. In static testing, codes (symbols with details) are checked while the model is being built incrementally to ensure that the right data and logic have been modelled. Such a static testing is preformed with the use of debugging facilities provided by the software or with comprehensive code check by modellers themselves, some other experts or non-technical person who is knowledgeable with the system. In dynamic testing, the behaviour of elements is checked by running sub-model or section of the model with insertion of additional codes (model instrumentation). The techniques used for this purpose are tracing, visual checks through animation, comparing output with expected or real-world results etc. in different situations or by forcing such situations. The project team needs to select the appropriate technique, depending on the prevailing situation. Use of different techniques portray that the different set of details are involved in. For example, walk-through method may not present details about the verification process and the output. In contrast, the comparison of model output with expected results may encounter much more details. Such varying degree of details can only be presented by referencing those in appropriate detail descriptions.

Significance: If the document delivers what elements and how the individual elements are verified, such records evidence that the model is verified and executes as intended.

Structure & Guidelines: Form 5.27 with table format is suggested as the key contents of the record has relative link from one set of details to another. Details are added by accumulating rows. Keys are suggested for convenience of recording.

MODEL LEVEL	ELEMENT TYPE	ELEMENT NAME	VERIFICATION TECHNIQUES EMPLOYED	VERIFICATION PROCESS & OUTPUT DETAILS
<i>State the name of the mode level.</i>	<i>State the name of the common element type.</i>	<i>State element and data item verified.</i>	<i>State the technique(s) employed for verification.</i>	<i>State briefly the verification process and out put, referring in the description for recording more details as appendix pages.</i>
* KEYS	Techniques		References/Appendix	
	1.	Walk-through with an experts		
	2.	Visual check with system experts.		
	3.	Comparison with expected results.		
	4.	(Define)		

Form 5.27 : Verification Details

Feedback to Previous Task Documents: Failing to verify model elements or data makes the changes to 'model details' task document.

5.3.5 Details in Operational Validity (B.3.2.1)

Details in operational validity are the major constituent in model validation process that ensures the creditability and confidence of the model and model output. Similar to the verification process, the choice of subjective or objective validation techniques depends on the appropriateness and the preferences of the project team. However, the major attribute affecting the choice of the techniques is whether the system being modelled is observable (real) or non-observable. The depth of details involved in the validation process and the results depend on the technique used. For example, statistical tests and procedures are very common in objective approaches. They produce much detail in graphical and/or tabulated form. In contrast, in subjective approach, fewer details are involved due to its nature, like animation. From documentation point of view, the details that comprised of the validation process and the results, thus, vary with the approach. Regardless of the various types of approaches used, the project team should conclude the validation process by giving some assurance to the client with reasonable accuracy that the model corresponds to the real-world system or meet the customer expectation for a new system.

Significance: If the document delivers how the model is validated, what assurance is given about the working model, such details are important for all parties involved in the project, especially, for management and other decision makers to have confidence on model results.

Structure & Guidelines: Form 5.28 in table format is proposed to present related details. Further, the details may be provided as appendix and are accumulated by adding rows.

VALIDATION TECHNIQUES EMPLOYED	PROCESS & DATA DESCRIPTION IN VALIDATION	RESULTS OBTAINED AND CONCLUSION
<i>State the technique employed for validation.</i>	<i>Describe briefly the process and any data involve in validation.</i>	<i>Describe briefly the conclusion made based on the results obtained. (Any lengthy results may be recorded as appendix or end pages but referencing them in the description.)</i>
References/ Appendix :	<i>State any appendix or references made in the conclusions and results obtained and where those references are available.</i>	

Form 5.28 : Details in Operational Validity

Feedback to Previous Task Documents: Failure to validate the model may result to complete change over 'model elements and details' and the subsequent task documents.

5.3.6 Details in Parameter Validity (Sensitivity Analysis) (B.3.2.2)

Sensitivity analysis is generally defined as the systematic investigation of the reaction of the model outputs to drastic changes of the model input data. In other words, it primarily concerns with validation of the model for drastic or extreme changes of the estimated or inaccurate data. Therefore, in sensitivity analysis, the estimated or inaccurate data are

changed, probably with significant amount or extreme conditions, and then the model is run to see the effect of key results. If the key results are too sensitive, the data are required to re-acquire and further sensitive analysis is to be performed, until the key results are not sensitive with certain acceptable level. If all fails, the project team need to record the model outputs that are likely to affect due to data inaccuracies.

Significance: If the document delivers how the sensitivity analysis is performed, what assurance is given about the data accuracy, and any results that are likely to affect in failing to achieve the expected accuracy, such details are important for document user to understand the results and recommendations.

Structure & Guidelines: Form 5.29 in table format is proposed to present related details. Furthermore, details may be provided as appendix and are accumulated by adding rows.

DATA ITEM ANALYSED	BRIEF DESCRIPTION ON ANALYSIS PROCESS	RESULTS OBTAINED, CONCLUSION MADE, ANY RESULTS BE AFFECTED
<i>State the data item analysed.</i>	<i>Describe briefly the sensitive analysis process.</i>	<i>Describe briefly the conclusion made based on the results obtained; any output affected, etc. (Any lengthy results may be recorded as appendix, but referencing them.)</i>
References/ Appendix :	<i>State any appendix or references made in the conclusions and results obtained and where those references are available.</i>	

Form 5.29 : Details in Parameter Validity/Sensitivity Analysis

Feedback to Previous Task Documents: Failing to achieve the expected level of data accuracy forces to re-acquire the data and alter the details in ‘data description’.

5.4 Analysis of Task Documents in Phase 03: Experimentation

5.4.1 Overview on Experimentation (C.1.1.1)

The effort and time required for experimentation depend on many factors. They primarily depend on the behaviour of the model. For example, if the model is deterministic, several numbers of replications may not be required to acquire accurate results. However, majority of simulation models include some element of randomness. Therefore, the options available in designing and analysing simulation experiments are next influenced by the model-terminating situation. For example, if there is a natural terminating point to an experimental run; then less or no effort is required for deciding the run length, as the analysis and conclusions are based on the output values, produced at the terminating point. In contrast, in non-terminating simulation, the model could theoretically be run for an infinite of period or for a specified period, determined through a tactical plan. This is where the simulation state is important in deciding the run length. If the simulation is in steady-state cycle, its repeated cycles of equal length make the choice for the run length convergent. In contrast, steady-state situation depicts not only the necessity of much effort for deciding the run length, but

also concern about the warm-up period, before the model reach to the normal working condition. Once these characteristics are identified, methods of experimentation are decided, based on the purpose of the simulation project. Interactive method could be the case, if the purpose is for investigating specific conditions or training purpose. Similar situation may be found when the details are not necessary for comparative or predictive purpose and the time is not a constraint for the experimentation. In contrast, experimental factors and their level are required to set in batch experimentation for comparative or predictive purpose. Decisions on these experimental characteristics reflect the effort and time required for the entire experimentation process. For example, predictive experimentation requires more replication and produces details that are more precise.

Significance: The document that contains the model characteristics, model behaviour and experimentation approaches show the overall requirements in experiment design and run.

Structure & Guidelines: The contents of the task document may not have direct relationship to justify a table format, though they show some indirect relationship. However, Form 5.30 is proposed to record the background details of the experimentation process.

MODEL TYPE:			
True Deterministic	:	<input type="text" value="Yes/No"/>	Stochastic : <input type="text" value="Yes/No"/>
MODEL END SITUATION:			
Terminating	:	<input type="text" value="Yes/No"/>	Non-Terminating: <input type="text" value="Yes/No"/>
SIMULATION STATUS:			
Steady-State Cycle	:	<input type="text" value="Yes/No"/>	Steady-State : <input type="text" value="Yes/No"/>
EXPERIMENTATION APPROACH(ES):			
Interactive	:	<input type="text" value="Yes/No"/>	Batch : <input type="text" value="Yes/No"/>
Comparative	:	<input type="text" value="Yes/No"/>	Predictive : <input type="text" value="Yes/No"/>
ADDITIONAL NOTES	:	<input type="text" value="State additional notes, if available, to support the experimentation environment."/>	

Form 5.30 : Overview on Experimentation

Feedback to Previous Task Documents: There is no effect on previous task deliverables due to changes in details of record. However, the task documents in respect of experiment and report details may be required to update as a result of the analysis of experimentation.

5.4.2 Warm-Up Period (WUP) and Starting Conditions (C.1.2.1)

Estimation of WUP is primarily associated with the answer to the practical question “Does the state of the model represent the normal working conditions?” This can be answered by viewing the running model or analysing and inspecting the reports collected from the model run. The selection of the method for determining WUP depends on the statistical ability of the model executor or project team and the time constraint. Regardless of method selected, different replications with different set of random numbers produce different values for

WUP. Therefore, it is important to perform a minimum number of replications and select the longest WUP. Moreover, if the model has more than one experimental factor, WUP should, theoretically, be determined for each experiment. However, in reality or due to time restriction, it is very common that the same WUP is used for each experiment.

Initial starting condition of the model depends on the real system. Some model may start with empty (e.g. tool-store). However, many manufacturing systems start with some fixed or variable values. The starting condition never influences the steady state of the model. Instead, it converts the transient stage of the model into steady state with minimum or no time, resulting less time for experimentation. Some existing simulation software allows establishing the initial conditions. As the availability of real-data is the major limitation to the initial condition, the project team needs to decide WUP with or without them.

Significance: The document that contains the values of the starting condition, how the starting condition is established, how theoretical WUP designed, enables the document user to replicate the experimentation process accurately.

Structure & Guidelines: Form 5.31 is suggested to maintain the consistency of records. However, a need may arise to present the comprehensive details with few diagrams and tables, etc., referencing in the description.

STARTING CONDITION:			
Starting Condition	:	<i>State whether the starting conditions is zero or variable.</i>	
Data and Description	:	<i>If the initial starting condition is needed, describe briefly what and how key inputs are fed into the model.</i>	
WARM-UP PERIOD:			
Visual Inspection	:	<i>Yes/No</i>	Time Series Analysis : <i>Yes/No</i>
Theoretical Value & Units:	<i>Value & Units</i>		
Data and Description	:	<i>State description on key outputs, number of replications, process of estimation (in case of visual inspection), reasons on judgements, etc. In case of time series analysis, a need arises to present graphical content as appendix, but referring in the description.</i>	
References/Appendix	:	<i>State any references made in the description and where those references are available.</i>	

Form 5.31 : Warm-Up Period and Starting Condition

Feedback to Previous Task Documents: There is no effect on previous task deliverables due to changes in details of this record.

5.4.3 Run Length (C.1.2.2)

Establishing the run length of a simulation is concerned only in steady-state cycle and steady-state simulations, as the terminating simulation, by their nature, have a defined end event, which find the end of run length. Therefore, the number of replication is the only

effective sample increasing strategy available for those terminating model. However, in model associated with steady state, the run length is to be determined through event observation (rule of thumb) or graphical approach (time series analysis) similar to WUP. Event observation, in which the most infrequent events are considered to decide the run length, is straightforward. However, its inability to produce accurate results directs to adhere to an accurate approach (time series). In time series analysis, the results are collected from WUP point for a period of length, which is estimated through data observation, for a few numbers of replications (at least 3 in practice). Then, the results are plotted against the time to see how different set of results converges. Hence, the run length with certain confidence can be found. However, similar to WUP, a certain over-estimation is common.

Significance: The document that contains how the theoretical value for the run length was decided enables the document user to repeat the experimentation process accurately.

Structure & Guidelines: Form 5.32 is proposed by making a provision for extra details as appendix.

APPROX. VALUE & UNITS IN EVENT OBSERVATION	:	<i>Value & Units</i>	LIST OF INFREQUENT EVENTS	:	<i>Infrequent Event(s)</i>
THEORATI. VALUE & UNITS	:	<i>Value & Units</i>	% OF CONFIDENCE	:	<i>Value</i>
DATA AND DESCRIPTION	:	<i>State description on key outputs, number of replications, any assumptions, reasons on judgements, etc in run length estimation process. The details in determining theoretical values can be presented as appendix, but referencing them in the description.</i>			
References/Appendix	:	<i>State any references made in the description and where those references are available.</i>			

Form 5.32 : Run Length

Feedback to Previous Task Documents: There is no effect on previous task deliverables due to changes in details of this record.

5.4.4 Number of Replications (C.1.2.3)

Replication is the process of executing the same simulation experiment over a number of times with different random number streams in stochastic process. The number of replications directly depends on the characteristics of the experimentation approach. For example, many replications may not be performed in interactive simulation due to time restriction. In batch experimentation, if the results are obtained for comparative purpose, less number of replications may be sufficient. In contrast, more replications are required for accurate predictive experimentation. It is generally accepted that 3-5 replications are performed to maintain accuracy of the results. However, the graphical approach, similar to

WUP and length of run, provides much more feasible answer for number of replications. In graphical approach, the cumulative averages of key measures of performance are generally plotted against the number of replications. The number of replications is determined with confidence at the point where there are no much changes in cumulative values. Theoretically, the number of replications varies with the experiments. However, in practice, it is very common that fixed number of replications is used for each experiment.

Significance: The document that contains details about theoretical values for number of replications enables the document user to imitate experimentation accurately.

Structure & Guidelines: Form 5.33 with provision for extra details as an appendix is suggested.

THEORATI. VALUE & UNITS	:	<input type="text" value="Value & Units"/>	% OF CONFIDENCE	:	<input type="text" value="Value"/>
DATA AND DESCRIPTION	:	<i>State description on key outputs, assumptions, etc in estimating No. of replications. With graphical approach analysis, a need arises to present graphical contents as appendix, but referencing them in the description.</i>			
References/Appendix	:	<i>State any references made in the description and where those references are available.</i>			

Form 5.33 : Number of Replications

Feedback to Previous Task Documents: There is no effect on previous task deliverables due to changes in details of the record.

5.4.5 Summary of Experiment Run (C.2.1.1)

The task documents in design of experiments sub-phase present the theoretical aspects in the experimentation process. Those details provide the base information in experimentation. But, in-reality, many other factors such as the time, resources, and costs are considered for deciding the values to be practiced. For example, theoretically, WUP, run length and number of replications should be determined for each experiment. However, in practice, it is very common that certain over estimated fixed values to those theoretical values are considered during the experimentation process, depending on the constraint. Those decisions may have a certain influence on the results, conclusions, and recommendations. Furthermore, if one or more experimental factors with many levels are available, then, theoretically, there is a potential for a large number of experiments to be performed. But, in reality, a limited number of levels are considered for experimentation, depending on the constraints. Conclusions are then drawn considering the likely out comes through statistical methods. Moreover, preliminary experiments may be performed before the real experiments begin for the purpose of measuring the effect of certain experimental factors.

Significance: The document, which contains details about the experiments conducted, their factor levels, if available, experiment approach and the practiced values on designed parameters, helps the document user to be aware the way that the experiments were conducted, and the constraints affected during the experimentation. Hence, the user enables to analyse likely outcomes, before the conclusions and recommendations are made.

Structure & Guidelines: Form 5.34 with table format is suggested as the key contents of the record has relative link from one set of details to another.

EXPERIMENT NUMBER	EXPERIMENT DETAILS	NO. OF FACTOR LEVELS PRACTICED	PRACTICED WARM-UP PERIOD & UNITS	PRACTICED RUN LENGTH & UNITS	NO. OF REPLICATIONS PRACTICED	EXPERIMENT APPROACH	USEFUL NOTES
State experiment number or ID or name conducted.	State individual or common description on experiments conducted, depending on subsequent columns.	State the practiced No. of factor levels of experiments specified in column 1, if available.	State the practiced warm-up period of the experiments specified in column 1.	State the practiced run length of the experiments specified in column 1.	State the practiced No. of replications of the experiments specified in column 1.	State the experimentation strategy (Interactive, predictive/comparative) adopted for experiments	State any reasons on the decisions made regarding the details in previous columns or any other useful notes.

Form 5.34 : Summary of Experiment Run

Feedback to Previous Task Documents: A need may arise to update the experimental and report details that have been decided at the beginning of the project.

5.4.6 Summary of Results (C.2.2.1)

In simulation, many experiments are conducted to obtain various results. Results are then analysed while the problem objectives are being kept in attention. However, the choice of input variables, the results obtained and the methods adopted for results analysis are influenced by various combinations of many factors such as the purpose of the project, the facilities in the simulation software, the skills of the project team and the preference of each individual. Therefore, a generic form to record the results or the process of results analysis is not proposed or is not even important. Instead, only the significant results, which will influence the conclusions and the recommendations, can be summarized against each or set of experiments performed. For example, when multiple replications are performed for one level of experiment factor, whilst the other factors are keeping in certain fixed level, if available, individual results in each replication or average values for a range of replications can be recorded against the experimental factor. However, provisions can be made for enclosing the significant results in tabular and graphical form for each experiment conducted and for result analysis process as additional information.

Significance: If a summary of the results of experiments or the outputs of result analysis process are recorded clearly and concisely against the experiments, such records enable the document user (particularly the client) to review the final results, the alternatives that have been addressed, the criterion by which the alternative solutions are compared and the conclusions and recommendations made by the analyst.

Structure & Guidelines: Form 5.35 with table format is proposed to represent related details. The details are accumulated by adding relevant rows. It may be necessary to alter number of columns to suit different factor levels.

EXPERIMENT								RESULTS			
NO.	FACTOR 1		FACTOR i		FACTOR n		REP. NO. /NO. RANGE	State PM Name-1	State PM Name-i	State PM Name-n	App/ Ref.
	NAME	VALUE	NAME	VALUE	NAME	VALUE					
State the experiment number.	State the name of the experiment factor "1".	State the value of the experiment factor "1".	State the name of the experiment factor "i".	State the value of the experiment factor "i".	State the name of the experiment factor "n".	State the value of the experiment factor "n".	State the replication number or number range.	State the individual/ average value of the results "1".	State the individual/ average value of the results "i".	State the individual/ average value of the results "n".	State appendix name of graphical/Tabular results

Form 5.35 : Summary of Results

Feedback to Previous Task Documents: Result analysis may lead to the changes in experiments run and subsequent task documents, in case of failure to obtain useful results.

5.4.7 Conclusions and Recommendations (C.2.2.2)

One of the final steps in any decision making process is to make conclusions and recommendations. The conclusions are drawn from the results and other interpretations obtained from result analysis process. Recommendations are then made, based on the conclusions. However, there are no specific guidelines on how the conclusions are drawn or how the recommendations are made. Such analysis solely depends on the project, need of the client as well as the skills of the project team. However, one thing is certain that the conclusions are based on different set of experiments results and their analysis. Therefore, the conclusions may summarize different sets of alternative solutions to the problem studied. This demands that the cost/benefit ratios are to be investigated for each alternative, before any recommendation is made. Therefore, the recommendation is based on the conclusions drawn previously and some other factors associated with the particular problem or the organization. However, this iterative process may lead to further experimentation to ensure that the problem objectives are achieved.

Significance: If the document delivers all inferences drawn and recommendations made with details associated in reckoning process, such details are important for all parties involved in the project, especially, for management and other decision makers to aware the benefits and the success of the project.

Structure & Guidelines: Form 5.36 is proposed for recording details in free-body text format. However, it is recommended to accommodate the structure and guidelines shown in the form to maintain the uniformity of the records.

CONCLUSIONS :	
No.	State the conclusion(s). <ul style="list-style-type: none"> ○ Based on which experiment(s) (perhaps, the numbers stated in the summary of results) and which results.
RECOMMENDATIONS:	
No.	State the recommendation(s). <ul style="list-style-type: none"> ○ Based on which conclusion(s) ○ Based on (State other factors).
OTHER DETAILS :	
State any other important details influenced on the above decisions, if available, or for better understanding of the above conclusions and recommendations.	

Form 5.36 : Conclusions and Recommendations

Feedback to Previous Task Documents: A need may arise to update the summary of experiments run and the subsequent task documents, in case of further experimentations.

5.5 Analysis of Task Documents in Phase 04: Project Implementation

5.5.1 Review of the Results (D.1.1.1)

Results are communicated by means of written reports, formal presentations, and demonstration of the working model. Selection of the medium depends on the needs for the project and the participants in the reviewing process. However, reporting or presenting simulation results alone may not fulfil the purpose of this task. It is also essential to deliver a summary of the project, which may be extracted from previous task documents such as problem objectives, experiment details, model summary, etc., for better understanding of the results communicated. The details that are to be delivered for this purpose again depend on the participants and their level of contribution made for or involvement with the project. During this evaluation process, they may not agree, completely agree or partially agree with the recommendations proposed. If they do not agree, they may be able to suggest alternative approaches for addressing the issues. Such situation may create further experimentations.

Significance: The records that deliver how the results were evaluated and what feedback was received from the participants help the document user to be aware the level of agreement of results and recommendations with the client.

Structure & Guidelines: Form 5.37 is proposed for recording details in free-body text.

WRITTEN REPORTS :	<input type="text" value="Yes/No"/>	PRESENTATION AND/OR DEMONSTRATION :	<input type="text" value="Yes/No"/>
PARTICIPANTS FOR REVIEW :	<input type="text" value="State to whom the details were presented or reported (E.g.: Team member(s), clients, implementation team etc.)."/>		
LIST OF DETAILS REPORTED OR PRESENTED :	<input type="text" value="State the list of details reported, presented or demonstrated (e.g. objectives, summary of results, recommendations, model demonstration, etc.)."/>		
RECEIVED FEEDBACK :	<input type="text" value="Yes/No"/>	AGREED LEVEL OF RECOMMENDATIONS :	<input type="text" value="Yes/No/Partly"/>
ANY ALTERATIONS TO RECOMMENDATIONS :	<input type="text" value="If the recommendations are not agreed or partly agreed, then present any alterations to recommendations or any new ideas presented together with reasons for such alterations."/>		

Form 5.37 : Review of the Results

Feedback to Previous Task Documents: Failing to agree the recommendations fully or partially may result to changes in summary of the experiment run and the subsequent task documents.

5.5.2 Review of the Project (D.1.2.1)

In review of any project, the basic question should be answered is whether the target or the objectives of the project have been met. Recording those objectives, which have been met or not met together with any reasons for failing to achieve, give an overall assessment about the performance of the project. At the end of the project, the project team can present a self-assessment about their work in respect of overall performance, any specific phases, sub-phases, or task performed well and should have been performed better together with any reason behind for not performing well. They can summarize their assessment stating any particular areas that need further improvement in order to ascertain better results. Similar way, the client, or the management can present their views in respect of satisfaction level of the overall project management (e.g. managing time scale), what the project have achieved and what should have been achieved. In case of failure to present such an assessment due to some reasons, for instance, too early to assess without implementing the project, additional notes could be presented for future references. Such provision may be more useful, if the project is not yet implemented. However, in rare instances, the project review may not be able to furnish until the implementation is over.

Significance: The records that differentiate the problem objectives met from those have not yet met and the perspectives of the project team and the client about the project help the document user to evaluate the project and to make improvements, before reusing the model.

Structure & Guidelines: Form 5.38 is proposed for recording details in free-body text format. The guidelines provided inside the form give more information about the contents.

STATUS OF THE PROBLEM OBJECTIVES :	
Objectives, met :	<i>State the problem objectives that were met through experimentation (before implementation).</i>
Objectives, not met :	<i>State the problem objectives that were not met through experimentation (before implementation) and the possible reason for such pitfalls.</i>
PERSPECTIVES OF THE PROJECT TEAM :	
Overall performance :	<i>Fully Satisfied/Moderately Satisfied/Satisfied/Unsatisfied</i>
What has been done well? :	<i>State any areas, sub-phases or phases that were done well in the project.</i>
What should have been done better? :	<i>State any processes or actions that should have done better, but could not perform well due to certain reasons.</i>
What needs to be improved later? :	<i>State any processes, actions, areas, sections or phases that need further improvement for better results.</i>
PERSPECTIVES OF THE CLIENT :	
Overall Satisfaction :	<i>Fully Satisfied/Moderately Satisfied/Satisfied/Unsatisfied</i>
What has been achieved well? :	<i>State what achieved well within the constraint available for project or project team.</i>
What should have been achieved better?:	<i>State what have not achieved up to the level of expectation and brief the reason for such pitfalls in consultation with the team.</i>
ADDITIONAL NOTES :	<i>State any additional notes that are missed out above, in respect of positive (mostly) or negative comments.</i>

Fig. 5.38 : Review of the Project

Feedback to Previous Task Documents: There is no effect on previous task deliverables due to changes in details of record.

5.5.3 Implementation Plan (D.2.1.1)

Project implementation team may be the simulations team itself or a completely outside team or a joint team. In case of an outside team, it is the responsibility of simulation team to ensure that they have understood the recommendations fully. But, the proved compromise is the joint team, which comprise of one or a few members of the simulation team with outside members for better implementation. Regardless of the team, the recommendation proposed may not always be implemented in the same way due to outside issues such financial, time and resource constraints. This in turns a minor adjustment to the proposed recommendations or a proposal for different implementing stages. For instance, first resource can be added to the system in first stage and the second one in the next stage. Therefore, the recommendations and their implementation plan with time scale for each stage are to be identified and recorded clearly so that a mechanism could be set out for monitoring the progress in the implementation process. Such records enable the implementation team to evaluate the model and make any alterations, particularly, in case of faulty implementation and/or change in the system environment, hence, to ensure the success of the project.

Significance: The records, which deliver who will implement the project, what and how the recommendations are practiced and in what stages that the progress should be monitored, help the document user to see how the implementation plan is performed.

Structure & Guidelines: Form 5.39 with some tabular format is proposed for recording details in free-body text format.

RECOMMENDATIONS TO BE PRACTICED :		
<i>Recom. No.</i>	<i>Description of the recommendations to be practiced (may have slight adjustments to the recommendation made at the end of reviewing result task).</i>	
MEMBERS OF IMPLEMENTATION TEAM :		
<i>State member list implementation group from;</i> <ul style="list-style-type: none"> ○ <i>Simulation team, and/or</i> ○ <i>Out side party.</i> 		
IMPLEMENTATION :	Stages	Completion/Due Date
	<i>State the implementation stages or complete implementation expected to be finished.</i>	<i>State the expected completion date(s) of full implementation or implementation stages.</i>
MONITORING :	<i>State the monitoring stage(s) (may be end of implementation stages or full implementation).</i>	<i>State the expected reviewing date(s) of monitoring stage(s).</i>

Fig. 5.39 : Implementation Plan

Feedback to Previous Task Documents: Failure to achieve the objectives or changes in the system environment may result to changes in 'summary of experiment run' and the subsequent task documents.

5.5.4 Documents Prepared (D.2.2.1)

The documents are prepared, by accumulating and arranging appropriate task documents that have already been prepared and stored during the simulation project, for different audiences in respect of maintaining the project, protecting the contingency losses, and enhancing the model reuse. Apart from the main audience - model re-users and other interested parties – targeted at this stage, the implementation team, if different, is also benefited from such documents. However, enclosure of different task documents may vary with the individual projects and their current status. Regardless of the task documents included, from documentation point of view, presenting details on different categories of documents prepared and making provision to access whole MRD database, enable the document user to extract details with his/her own choice in more expanded environment.

Significance: If the task document shows the status of the project at the time of preparation of documents, where the recorded details are stored, for what purpose and for whom documents were prepared, such records help the document user to trace the necessary information and study the report extensively.

Structure & Guidelines: Form 5.40 with some tabular format is proposed for recording details. Details of the documents are accumulated by adding rows.

DOCUMENT FILE NAME (S) :	<i>Document file name.</i>	CURRENT STATUS OF THE PROJECT :	<i>State the current status of the project (e.g. project implementation is underway).</i>
DOCUMENTS PREPARED :	Audience <i>State for whom (audience) each document is prepared.</i>	Purpose <i>State the purpose of each document prepared.</i>	

Form 5.40 : Documents Prepared

Feedback to Previous Task Documents: There is no effect on previous task deliverables.

5.6 Summary

This chapter presented the structure for presenting the details in each task document. An attempt has been made to place a generic structure and common layout for each task document using form approach. The use of generic or standard terms, the uniformity and consistency of records have been considered as significant factors in a standard documentation methodology. Due to the iterative nature of the simulation project, the potential feed back loop for each task document was examined, so that such affiliation could be represented in MRD framework and user interface, which will be portrayed in Chapter 06.

6. MODEL REPRESENTATION AND DOCUMENTATION (MRD) FRAMEWORK AND TOOL

6.1 Introduction

In Chapters 04 and 05, the details of a typical simulation project were explored and the structure for maintaining the uniform and consistent records was proposed for each task document. The task documents may have to handle with different software tools, depending on the nature of the details. Therefore, it is now discussed, how such records are managed in concurrent with model development process. In task documentation process, the details are required to share or exchange to and from the simulation software being used for modelling as well as other software being used for task documentation. Therefore, the user interfaces, which are developed to manipulate the functions in respect of data storing, retrieving, exchanging, and output preparation in different software, are exemplified briefly, emphasizing the key areas in MRD Tool developed for on-line documentation. However, the task related to model representation may take much proportion in the discussion, as it is the vital component in the integrated MRD system. In appropriate points in the discussion, characteristics of the MRD system are explored for a standard and integrated MRD system.

6.2 MRD Process

The proposed approach to MRD process is the task-oriented approach, with which each task document of primary tasks in a simulation project is recorded with the structure proposed and the guidelines provided in Chapter 5, while the activities in the project are being performed sequentially and/or iteratively. This concurrent process is depicted in Fig. 6.1.

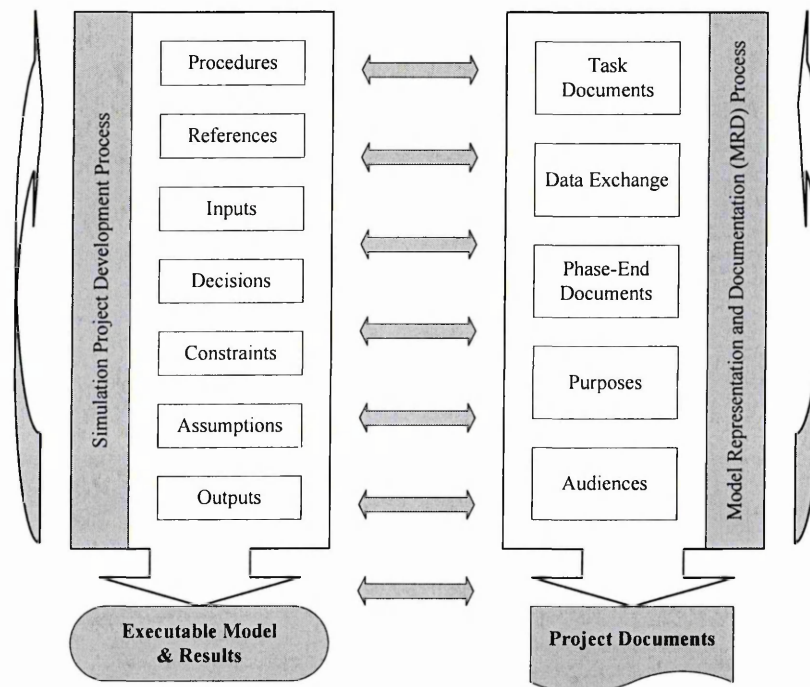


Fig. 6.1 : Overview of MRD Process

As depicted in the figure, the simulation model development process associates with details related to the procedures, references, inputs, outputs, constraints, assumptions, and decisions. While the model development process is being performed, in MRD process, such details are explored and extracted through the task documents for producing the phase-end documents for different purposes and different audiences. Some details extracted may have to exchange or share among the task documents as well as to and from the simulation software. Such an approach not only makes the MRD process un-isolated with the model development but also benefits for both processes mutually. Hence, this produces the implemented model with results and project document simultaneously as end results.

6.3 MRD System

MRD system is broadly and primarily for recording, and retrieving the details of primary tasks of the simulation project as the project progresses. As illustrated in Fig. 6.2, the details in each phase of the simulation project can be considered as repositories in different subsystems for generating the relevant phase-end documents. The details in different task documents are considered in different formats - text, data, or graphics -, depending on their nature as well as needs of sharing data among the other task documents and other software. For example, the project objectives, assumptions etc. are recorded in text format. Model data such as process times and resource schedules are recorded in data format so that they could be shared with the executable model. However, certain text records such as the descriptions of the model layout can be used in subsequent tasks like in model representation. Therefore, the structure of the text records plays an imperative role in data sharing or exchange function. The description in the model representation task produces both graphical and data contents for exchanging the details with other systems.

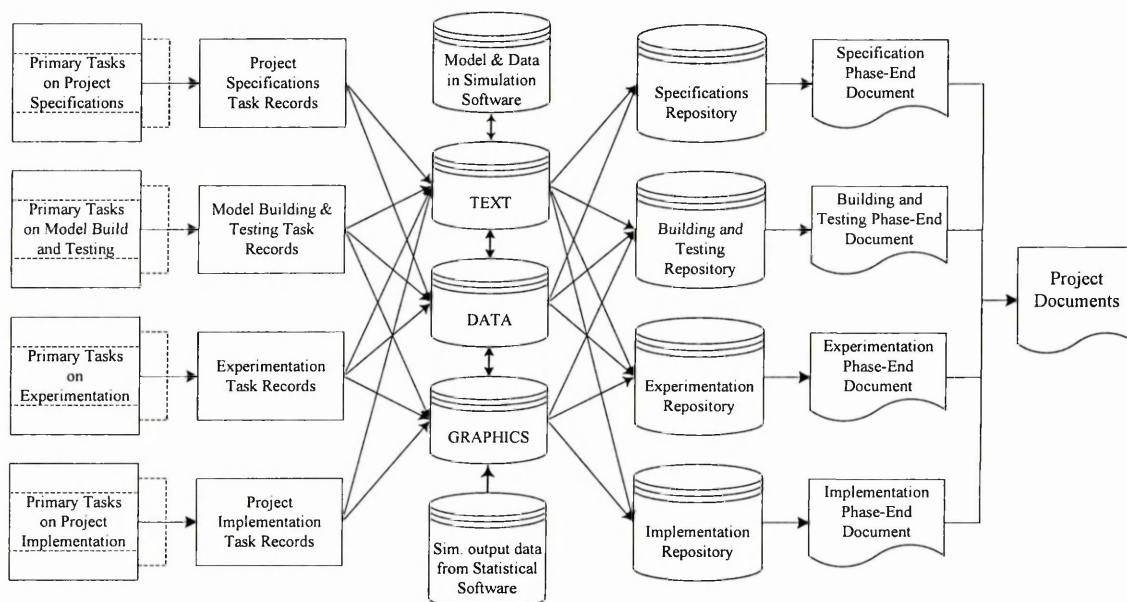


Fig. 6.2 : MRD System

Any need of data sharing reveals that the MRD system requires facilities for both functional and hierarchical integration of subsystems. Functional integration allows one subsystem (e.g. specifications repository) to share the data with other subsystems (e.g. design and testing data). Hierarchical communication permits the same data to share in different levels within the subsystem. Such communication between subsystems allows the document developer to produce documents needed at different stage – phase-end documentation. Composite of subsystems extends the production of documents completely for different purposes of different audiences. However, interfaces are needed not only for functional and hierarchical integration, but also for management of the task documents and exchanging the model and data with the simulation software being used. Together with such interfaces, MRD system will become an automated (rather semi-automated) MRD system.

6.4 Automated MRD System

Automated MRD system may be defined as a documentation management and control system that has a common layout and systematically linked details in each task of the simulation project, and that advises and helps the project team to not only document the simulation project, but also to perform the project with reasonable endeavour. For these purposes, an automated MRD system should endow with following capabilities:

- It should imitate the same procedure followed by the simulation project.
- It should handle task documents interactively to store and retrieve the details in sequential or interactive modelling process.
- It should provide the facilities for functional and hierarchical communication (for sharing details) among the task documents within and between subsystems.
- It should provide appropriate interfaces to translate documentation model, in some extent, to the simulation software that is intended to use (pre-documentation).
- It should extract the information on alteration made to the executable model so that the documentation model could be updated (post-documentation).
- It should facilitate to produce documents with a common layout for different purposes and different audiences in any stage of the project.

The above capabilities are accommodated through interfaces developed with VB and VBA. Interfaces facilitate to accommodate task documentation at any stage of the simulation project and to perform documentation as the project progresses. Interfaces necessary for the proposed MRD system are shown in Fig. 6.3. Due to diverse nature of the details, as illustrated in the figure, different software is necessary to capture them. In the proposed system, MS Word is primarily used for capturing and presenting details. However, the interfaces with Microsoft Access, MS Visio databases and simulation software (e.g. Arena) are necessary for an automated documentation system.

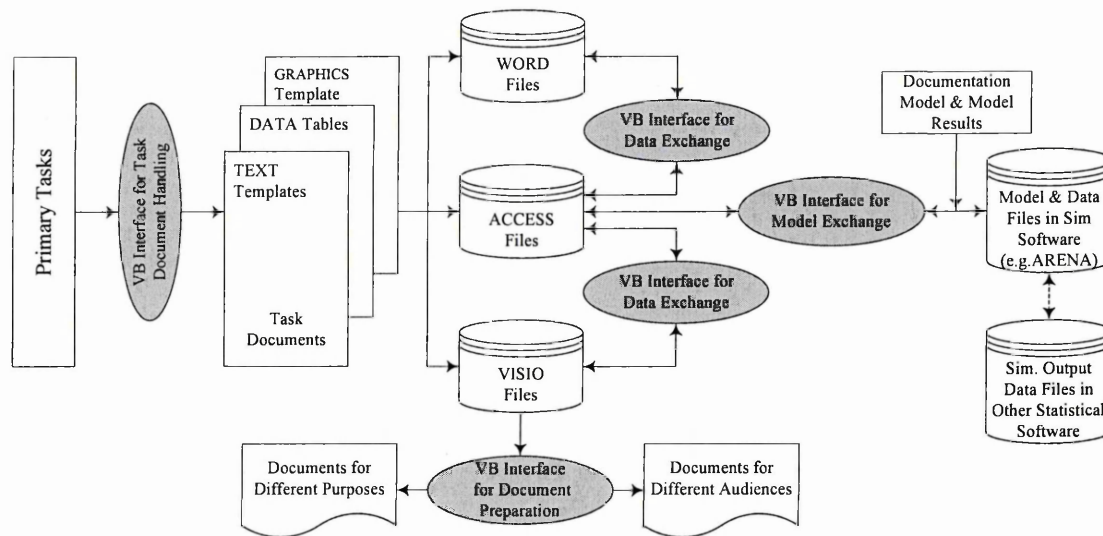


Fig. 6.3 : Automated MRD System

As depicted in Fig. 6.3, the task details are captured with the use of text templates, graphical templates, and data tables. Sequential or iterative tasks are managed by the 'interface for task document handling', facilitated to the user. With functional and hierarchical communication, the details are exchanged not only between the repositories among the different software used for documentation but also between the simulation software being used. Such data exchange is facilitated by the interfaces developed for 'data and model exchange' purpose. Once the task details are recorded, they are accumulated and arranged to produce the documents for different audiences for different purposes through the user interface developed for 'document preparation'.

6.4.1 MRD Subsystems

As stated previously, the records in each phase of the simulation project are considered as a subsystem. Subsystem is a document file that contains a collection of relevant task documents with different format of details. The files in each subsystem are the templates that restrain various forms, designed in Chapter 5, and are ready to collect details in consistent and uniform manner. The document writer allows using the template to record the details in different projects, interacting with other interfaced software and templates. Consequently, a subsystem comprises of the details that were resulted from the series of primary tasks, performed during the relevant phase of the simulation project. However, these records are either direct text entries to particular form with the proposed format or imported data tables or graphical objects from other software such as Access, Visio, simulation software, depending on the nature of record. This allows the document writer to handle the records in much easy environment and produce the documents as a text (Word) file. Fig. 6.4 illustrates the overview of the task details and their interaction between the MRD subsystems and the other interfaced MRD systems.

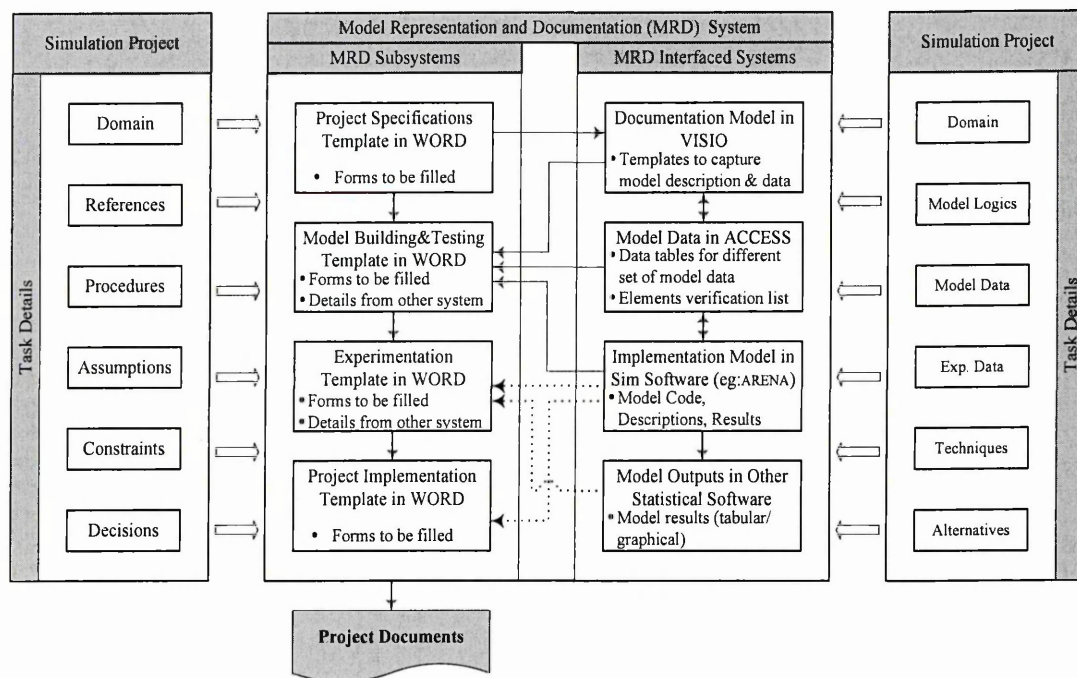


Fig. 6.4 : MRD Subsystems

Each form in Word templates has its own format to capture different details as direct input, or imported from previous task documents or from other interfaced systems. The structure of the forms is designed not only for presenting those details in a uniform manner, but also for facilitating for data sharing with the task documents and other interfaced systems. Blank spaces are left in templates for importing details (e.g. documentation model) from other interfaced sources. Although, Fig. 6.4 demonstrates the interaction of MRD subsystems with other interfaced system, it does not display how different details interact within the subsystem and what details are shared or exchanged with other systems. This indicates a need for exploring individual subsystems.

6.4.1.1 Project Specification Subsystem

Fig. 6.5 displays the task documents associated with the project specification template and the details shared with other interfaced software. The details in project specification subsystem file are mostly direct inputs from the planning stage of simulation projects. Hence, there is no much interaction with other software used for documentation. Although, no interfaces exist for importing details from other system, some details in task documents can be exported into next level of the subsystem - model building and testing. As illustrated in Fig. 6.5, one of such task document is layout detail that contains description about sub-models and resource elements. Those details are exported into Visio for creating the documentation model and subsequently for the implementation model. Such a provision provides more opportunity for the project team to share the same data for subsequent tasks in the simulation project, and thus to perform task documents as the project progresses.

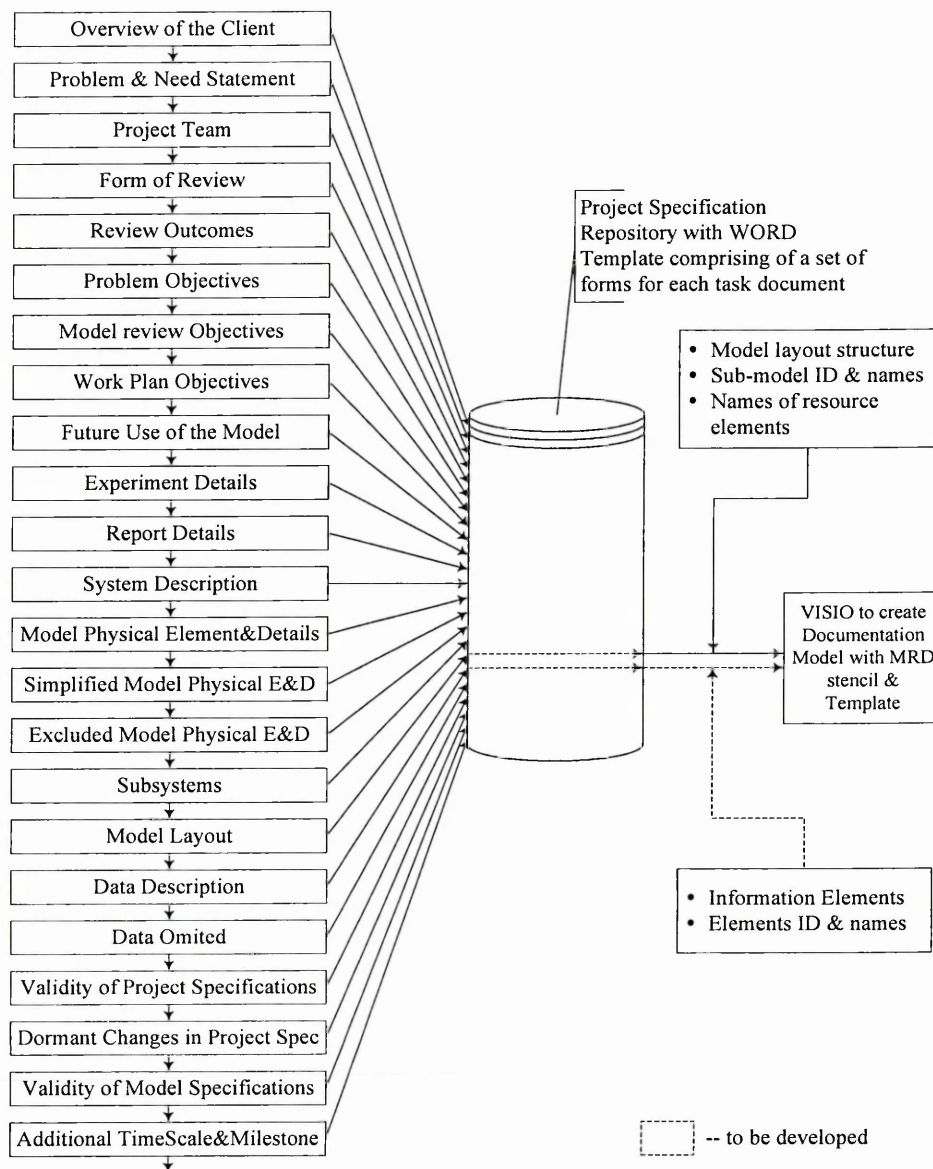


Fig. 6.5 : Project Specification Subsystem

6.4.1.2 Model Building and Testing Subsystem

Similar to the project specification subsystem, the model building and testing subsystem file is a Word template. However, most details for this file are imported from other software – Visio, Access and simulation software (e.g. Arena). The details in model descriptions and the model data are captured through Visio and interfaced MS Access, for which different sets of input data are required, as illustrated in Fig. 6.6. In other words, the details such as model logic, model data are not input to the Word template. Instead, they are the input to the documentation model that is generated by the use of the Visio template and stencil. The stencil comprises of a set of generic reusable model elements as graphical symbols as discussed in Section 5.3.1. However, layout data that have already been recorded in the project specification subsystem can still be used as input to documentation model. The elements in stencil library allow the document developer to not only represent the model

and capture model data, but also to lay a foundation to translate the model description into the simulation software. Once the model data are captured in Visio environment, they are then presented as Access tables with an appropriate format for recording them with the MS Word file. The details associated with the verification and validation tasks are the direct text inputs to the MS Word file. However, the list of elements for verification can still be generated through Visio and Access interfaces in order to maintain the uniformity of details.

Unlike in the project specification template, the task documents exemplified in Fig. 6.6 are required to be interfaced with other software to capture the relevant details completely or partially. Therefore, the model building and testing subsystem requires much user interaction for functional and hierarchical communication.

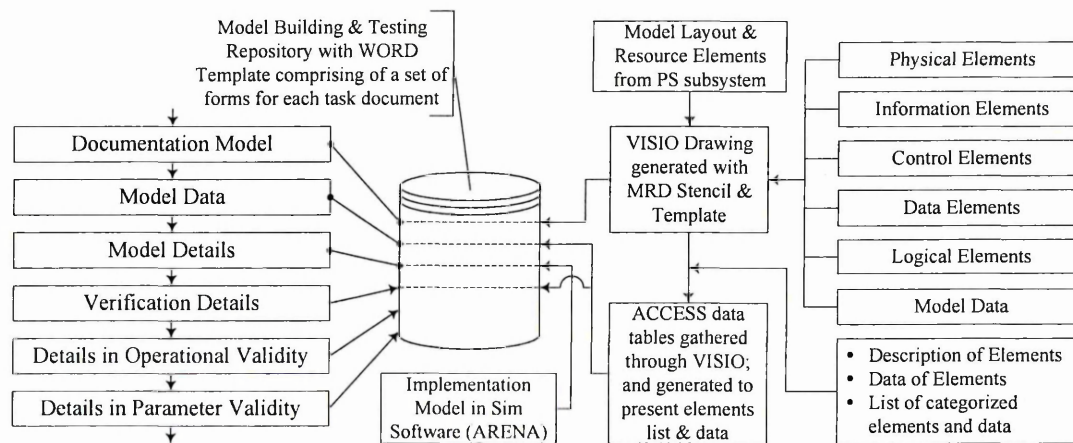


Fig. 6.6 : Model Building and Testing Subsystem

6.4.1.3 Experimentation Subsystem

The task documents in the experimentation phase, shown in Fig. 6.7, are recorded with the use of experimentation template. As shown in the figure, most of the details to be recorded are a combination of direct text entries to subsystem file and imported details from output database of the simulation software or other supporting statistical software. However, due to the differences in nature of output data (tabular or graphical forms), those details are not imported with direct interface (shown in faint dotted line) to the subsystem file. Instead, their summaries are suggested to present as direct text input to the subsystem file. Complete sets of details are proposed to place as appendix, but referencing them in appropriate forms in the subsystem file.

6.4.1.4 Implementation Subsystem

Implementation template comprises of four task documents as illustrated in Fig. 6.8. Details for this file are mostly direct text entries that are supplemented from model output database or the previous task documents. Therefore, no direct interface is necessary or proposed with other software for functional or hierarchical communication.

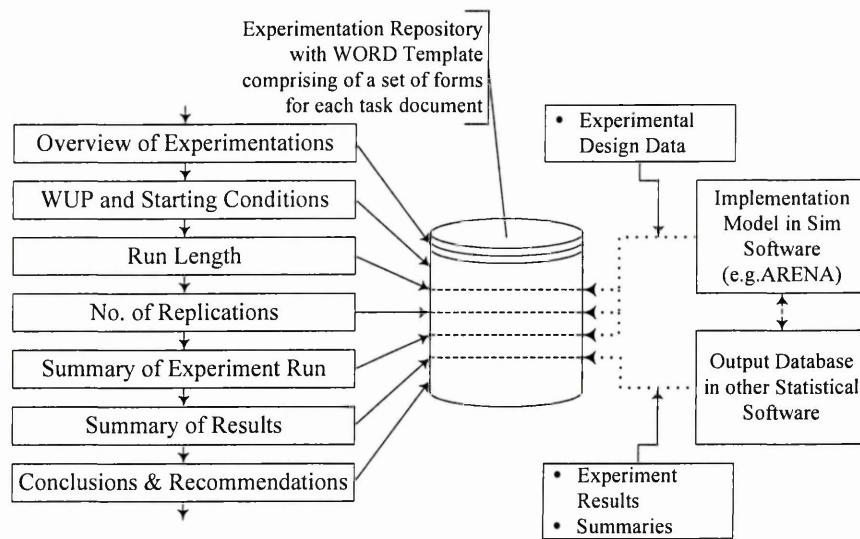


Fig. 6.7 : Experimentation Subsystem

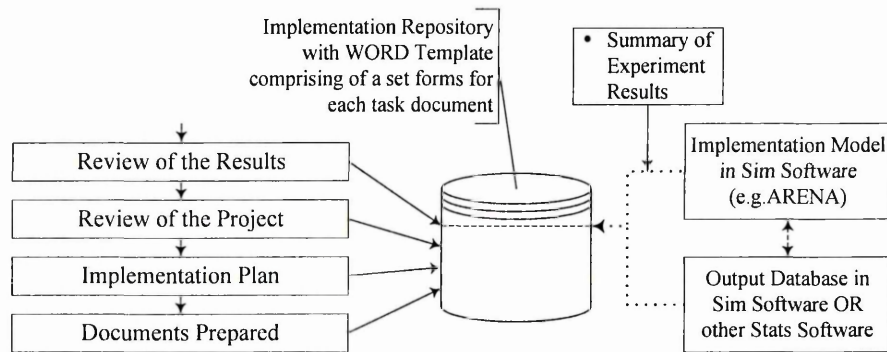


Fig. 6.8 : Implementation Subsystem

Brief discussion about the MRD subsystems portrays that the documentation model is significant for functional and hierarchical communication among the documentation subsystems, capturing model logic and data, and exchanging the model with the simulation software being used. Therefore, the documentation model is to be explored in-detail.

6.4.2 Documentation Model (DM) in the Core of MRD

Documentation Model (DM) may broadly be defined as a software independent graphical representation of the model. It primarily describes the interaction of reusable model elements graphically with the use of basic concepts in flowcharts technique. The graphical symbols relevant to the various types of elements in DM are mainly the customised basic symbols in flowcharts. In addition to the graphical representation, DM also captures the description of each model element and the model data, while the graphical DM is being constructed. Such a model description makes a provision to translate DM into the executable model in the simulation software. For these purposes, the symbols, which represent individual elements, are embedded with custom properties to capture different set

of information. The same information is used to produce a common database with a standard internal structure. Therefore, significant features in DM are the ability to represent or document the model independently from the software and the facility to create a common DM database for the benefit of other task documents as well as the executable model.

6.4.2.1 VISIO to Create DM and DM Database

As discussed the Chapters Four and Five, any simulation model may be described or documented by the use of physical, information (primary and auxiliary), control, logical and data elements. Constitutes in each element type may vary from one domain to another. However, the analysis of their behaviour emphasizes a need for establishing generic model elements so that they could be used to describe the model in distinguish domain and to create a common database to share details with other task documents and the executable model. The common Documentation Model database (DM database) is a collection of details of each element of the model. The details of each element are needed to capture and store in an element data sheet while the DM is being constructed. The data sheet may comprise of different fields to hold various types of data of the relevant element. The information in the data sheet can used to present the purpose and specifications of the elements for documentation purpose and to form a common DM database that enables model transfer with appropriate interfaces. Such capabilities are attained with the development of 'DM template' that comprise of a 'DM stencil' in MS Visio environment.

MS Visio is known for creating business and technical diagrams. It is a graphical package that document and organize complex ideas, processes, and systems. It is much popular in documentation of the software development process. Among many features, ability to create new stencils/templates, provision to capture attributes of individual elements, capability to generate diagrams through other sources, facility to link shapes/diagrams with external databases, potential on exporting individual shape data to generate a common database, automation capability through VBA, provision to publish diagram with other Microsoft software, and user friendliness makes the choice of Visio for creating the DM, and the common database in convenience and interactive environment.

6.4.2.2 DM Template

Documentation Model (DM) template is incorporated with predefined formats for presenting the details, and is linked with different databases to form a common database. The template is embedded with DM stencil that encompasses generic reusable model elements and other graphical symbols that are useful for documentation of simulation models in the context of flowcharts technique. Some of the model elements are interfaced with Access database to perform specific functions like capturing model data. The template

can be attached to construct DM in different projects. Model elements in DM stencil consider as masters and each master is embedded with its own custom properties to collect information for documentation and to form a common database. On drop of a master, customer properties screen is appeared to collect the relevant information from the user. The same information is applied for presenting the contents to be displayed in the element, just dropped. Such an approach enables the user to display the information in consistent and uniform manner. However, the user is still offered flexibility to alter the contents to be displayed with own choice. Such alterations do not change the properties in the data sheet and the DM database. Some of the elements may act more than one function. In such case, user is given a choice to select right function and input relevant information.

The example given in Fig. 6.9 shows the 'process' element (master - information type), what customer properties expected from the user, how customer properties are arranged to display the information of the element, and what options that user is given for selecting the appropriate function (process/delay) to suit the point of application.

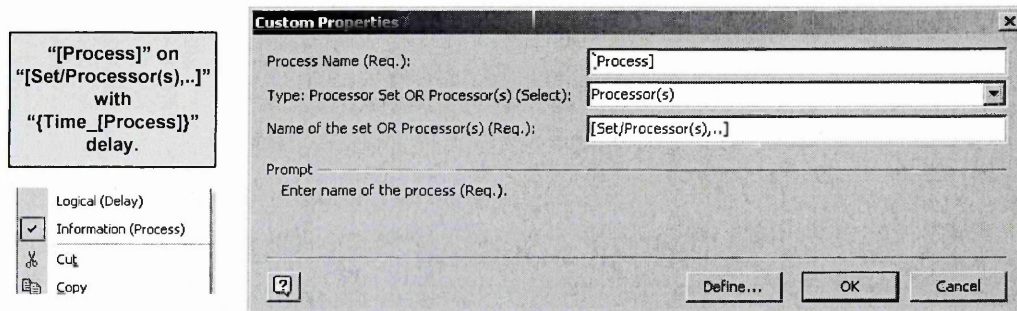


Fig. 6.9 : Process Element (Master)

It is clear from the above figure that the element is the information type and performs a process function (checked item). It captures a set or processor(s) specified in customer properties, delays the entity for a specified time {Time_ [Process]}, and frees the member of the set or processor(s) captured before. It is also visible that the user specified values in the custom properties window are appeared in the display of the element for documentation purpose. Inputs in the custom properties are stored in the element data sheet to create a common DM database. This option allows the user to produce uniform and consistent document, while the information are being collected for the executable model.

The text detail appeared in elements may not always be user inputs. For example, "Time_[Process]" represents identity for delay time, required for a particular process, and is a formed expression from user input to process name {[Process]}. From documentation point of view, this expression is used to identify and record process time in a data table. From modelling point of view, this statement is used as an expression of an attribute, assigned to an entity. The data values for attributes are extracted from the process time data

table, which is a component in 'model data' task document. Such an attempt to separate data from the model logic would benefit not only for documentation, but also for model construction. Model construction can be carried out with less interference with the model data (in case of none availability of data during the model construction) and the data can be updated in any time with less interaction of the model. Such isolation is much significant, if the data collection takes much proportion of the project time. However, data table may require a predefined specific format for model translation. Therefore, the user needs to present the data with a specified format. Table 6.1 shows the format of data required for process time in the proposed MRD methodology. The details presented in the table may not be essential for recording the model data. However, they are useful in assigning the values of attributes to relevant elements in model translation process.

PROCESS TIMES DATA						
Entity (Stage) Name	Associated Ele Type	Associated Ele Name	Expression	Expression Type	Data Values	Data Units
Entity A	Process	Turning	Time_Turning	Attribute	TRIA(1,4,8)	Minutes
Entity A	Process	Mill Loading	Time_Mill Loading	Attribute	EXPO(8)	Minutes
Entity A	Delay	Milling	Time_Milling	Attribute	TRIA(3,5,10)	Minutes
Entity A	Process	Inspection	Time_Inspection	Attribute	TRIA(1,3,4)	Minutes
Entity B	Process	Parting Off	Time_Parting Off	Attribute	UNIF(5,9)	Minutes
Entity B	Process	Grinding	Time_Grinding	Attribute	WEIB(10,1.5)	Minutes
Entity B	Process	Testing	Time_Testing	Attribute	TRIA(0.5,2,3)	Minutes
Entity B	Process	Inspection	Time_Inspection	Attribute	EXPO(3)	Minutes
ReGrindEntity A	Process	Grinding	Time_Grinding	Attribute	WEIB(6,1)	Minutes

Table 6.1 : Data Table for Process Time

6.4.2.3 DM Stencil

The symbols in the DM stencil belong to one of the main categories: physical, information (primary or auxiliary), control, logical and data elements and supporting symbols that are useful for documentation in the context flowchart technique. They are shown in Fig. 6.10.

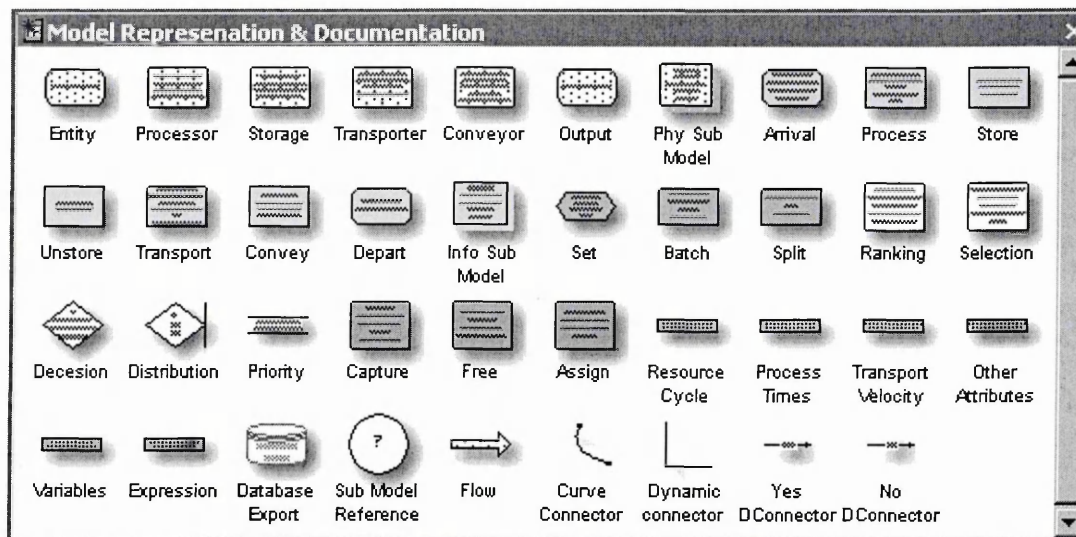


Fig. 6.10 : DM Stencil

Symbols for elements are differentiated with different shadings (and colours). In additions to shading (and colour) difference, individual element has its own shape to display the function of the element, but still maintaining the basic symbols in flowcharts. Different elements are embedded with their own customer properties that may be direct user inputs, derived from user inputs or some default values. Elements may also behave in different functionality, similar to ‘process’ element discussed previously. The customer properties and behaviour of all elements, proposed for the DM, are listed in Annexure A.

Brief discussion about the subsystem and their features clearly portray that the documentation needs a proper management of different software for capturing the various task details and for communicating task details in different functional and hierarchical level. This could only be achieved through well-developed interfaced on-line documentation tool.

6.5 MRD Tool for On-Line Documentation

MRD tool may be defined as a tool that guides and facilitates the document writer to produce project documents, while the project is being executed and immediate benefits from documentation are being delivered to the project. The MRD system and its subsystems with their interactions described previously highlight the specifications and interfaces to be incorporated in the MRD tool. The forms in task documents and its contents indicate a need for a development of a menu-driven documentation system. Such a system enables the user to call up the screen structures for the preparation of various types of documents.

The specifications for developing menus, screen formats, and user interaction should focus on providing options that can be selected by the user at any stage in the documentation process. With this view, the proposed main menu for the MRD tool consists of ‘project’ for handling the documentation process; ‘tool’ for interacting with the simulation software; and ‘help’ for providing brief instructions about the MRD process and the tool. Fig. 6.11 presents the main menu structure and menu items of the MRD tool, developed with VB. Table 6.2 shows the purposes or definitions of sub-menu items of each main menu.

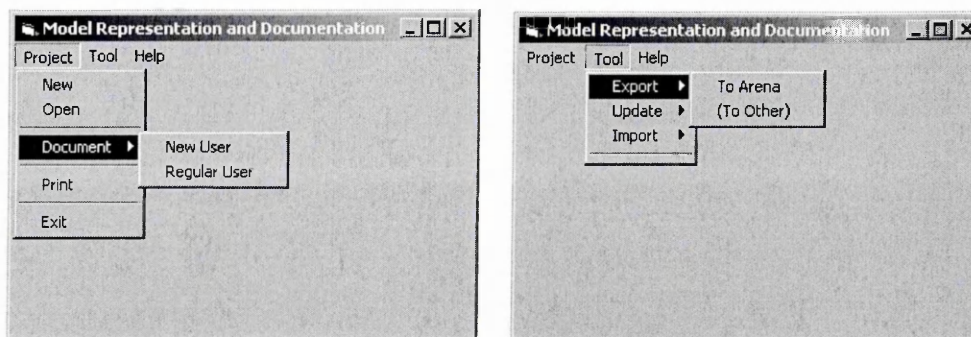


Fig. 6.11 : Menus in MRD Tool

Main Menu	Menu Item(1)	Menu Item(2)	Use in MRD Tool
Project	New	----	Allows establishing documentation environment for a new project.
	Open	----	Allows opening an existing document of a project.
	Document	New User	Allows documenting the project with expanded tree structure of the simulation project life cycle.
		Regular User	Allows documenting the project with direct tree structure of task documents of the simulation project life cycle.
	Print	----	Allows producing documents with the choice of the user.
	Exit	----	Exit the MRD tool.
Tool	Export	To Arena	Export the DM in Visio to Arena (Pre-Documentation).
	Update	From Arena	Allows updating (partially) DM in Visio with alteration made in Arena model (Post-documentation).
	Import	From Arena	This should allow creating complete DM from the existing Arena model. (To be developed.)
	(To Other/From Other)		*Need to develop interfaces for other simulation software.

Table 6.2 : MRD Menus

With above menus, it is expected that the project documents could be produced for different projects. The model details are exchanged between the documentation model and the executable model in the simulation software for making the documentation process un-isolated from the model development. However, the above menu structure does not facilitate for recoding of task details in task documents, for communicating between subsystems, and for handling of different software for recording different task documents. Such needs are managed with the user interface, developed for handling task documents.

6.5.1 Interface for Handling Task Documents

In the proposed on-line documentation tool, the user interface for handling task documents play a vital role for the entire documentation process. It provides the user to record task details in guided environment, interacting with different software and communicating with documentation subsystems. In specific terms, it should endow with following capabilities:

- It should reflect the different stages of the simulation project and relevant task documents so that any task documents could be accessed at any stage of the project, to accommodate sequential or iterative modelling process. Such facilities are accommodated by the expandable tree-view of task documents.
- It should distinguish between the task documents recorded and not recorded. This is achieved with the check boxes for task documents.
- It should indicate previous task documents that require attention or modification, in case of any alteration made to particular task document. This is attained with highlight task documents.
- It should visualize the task details immediately and allow accommodating alteration with least interaction. This is accomplished with the immediate screen window.

6.5.1.1 Tree View of Task Documents

The simulation project procedure is considered as highly iterative process. However, its sequential behaviour is considered for identifying the task documents, which are the accomplishments of each primary task. Task documents can be viewed as end branches of a tree - simulation project. Decomposition of project procedure into large and small branches and finally to task documents depicts the tree structure, as shown in Fig. 6.12.

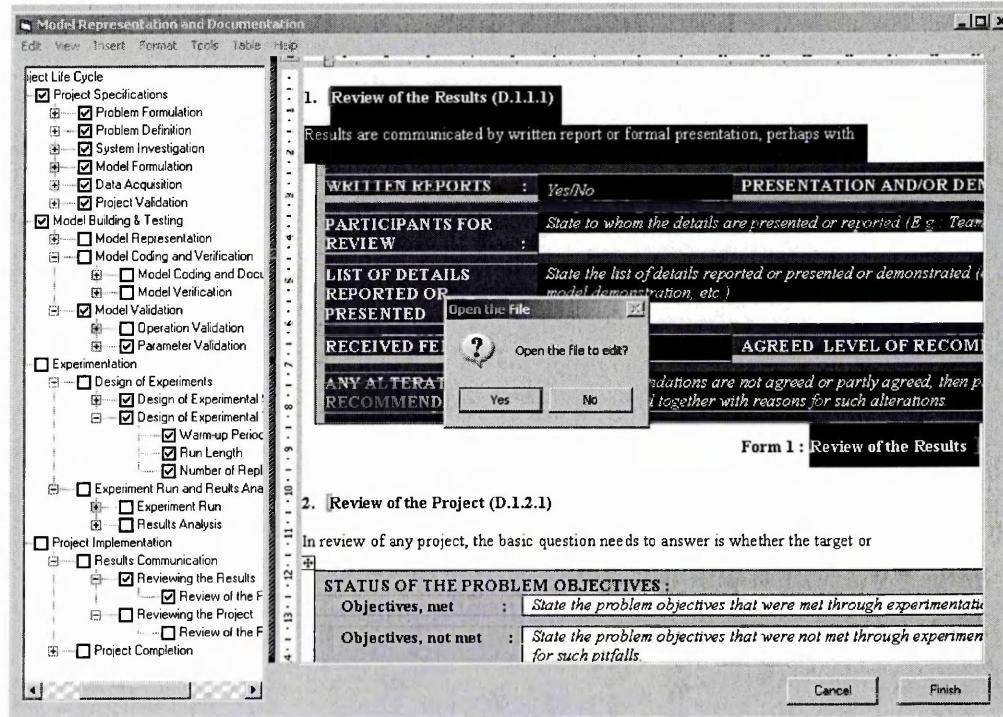


Fig. 6.12 : Tree View and Screen in Task Handling Interface for New User

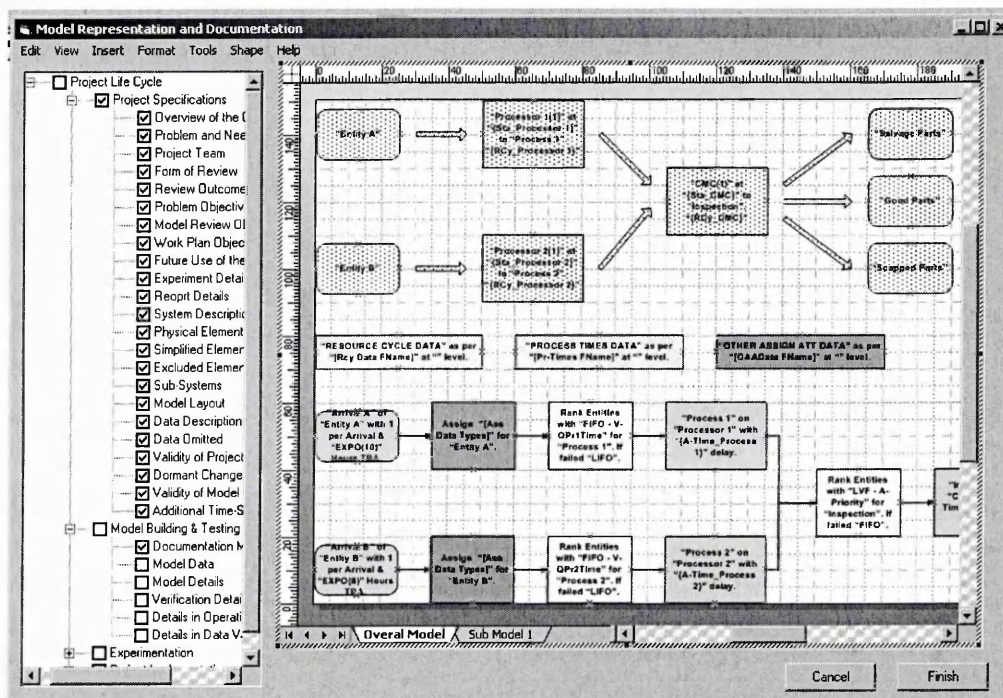


Fig. 6.13 : Tree View and Screen in Task Handling Interface for Regular User

Display of the entire structure of the simulation project may be an essential part, particularly for a 'new user' for task documentation process. However, as displayed in Fig. 6.12, stepping in each branch of the tree to reach fruits - task documents – would be tedious and cumbersome process. Therefore, an option should be available for 'regular user' to pick the relevant task document quickly, perhaps with potential indication with major phases of the simulation project, as illustrated in Fig. 6.13.

The end branches of the tree - task documents – act as a menu item to display the forms that are required to be filled or updated with task details. They also fulfil other potential interactive features such as sharing data with others task documents in the same subsystem or different subsystems; and communicating with other MRD software that are used to record different nature of details.

6.5.1.2 Check Boxes in Tree View

Due to iterative nature of simulation project procedure, the project team may intervene with ad-hoc stages in the simulation project. The same approach should reflect in the documentation process, provided that documentation and simulation are performed concurrently. Such ad-hoc attempts are displayed through check boxes that are accommodated with each branch of the tree view of task documents (see Fig. 6.12 and Fig. 6.13). However, the document developer should have the opportunity to mark task documents as recorded or unrecorded with his or her own choice.

6.5.1.3 Highlights of Task Documents in Tree View

It is a fact that simulation process is highly iterative. Adjustments made in any task or the relevant task document may raise a need for alterations to previous task documents. Deciding which task document(s) that needs alterations is crucial as it depends on what adjustments were made to the present task document. However, indicating potential task documents, that need attention, is worthwhile for the user to be aware with and to update them appropriately. Nevertheless, it is assumed that alterations in the subsequent task documents are obvious in a sequential process. Therefore, based on this assumption, the task document in the highest-level of the hierarchical tree is highlighted for the user to be aware with, which task document that needs alterations (see bold task documents in Fig. 6.12 and Fig. 6.13). Fig. 6.14 shows such potential task document in the highest level of the hierarchy in respect to the individual task documents.

6.5.1.4 Immediate Display Screen for Task Documents

Details that have been already recorded or need to be recorded in each task document are displayed in the immediate window, as in Fig. 6.12 and Fig. 6.13. With immediate display screen, the user could see the contents of task documents at any stage of the documentation

process with minimal interaction. The task document and the details, which are expected to record, act as a checklist for the task that is being performed. Therefore, such a display guides the user or project team to perform the relevant task with properly. The immediate window can accommodate minor alterations to task details with the menus offered with the particular software (Word and Visio). It also allows the user to access into the particular software environment for major adjustments with his or her choice.

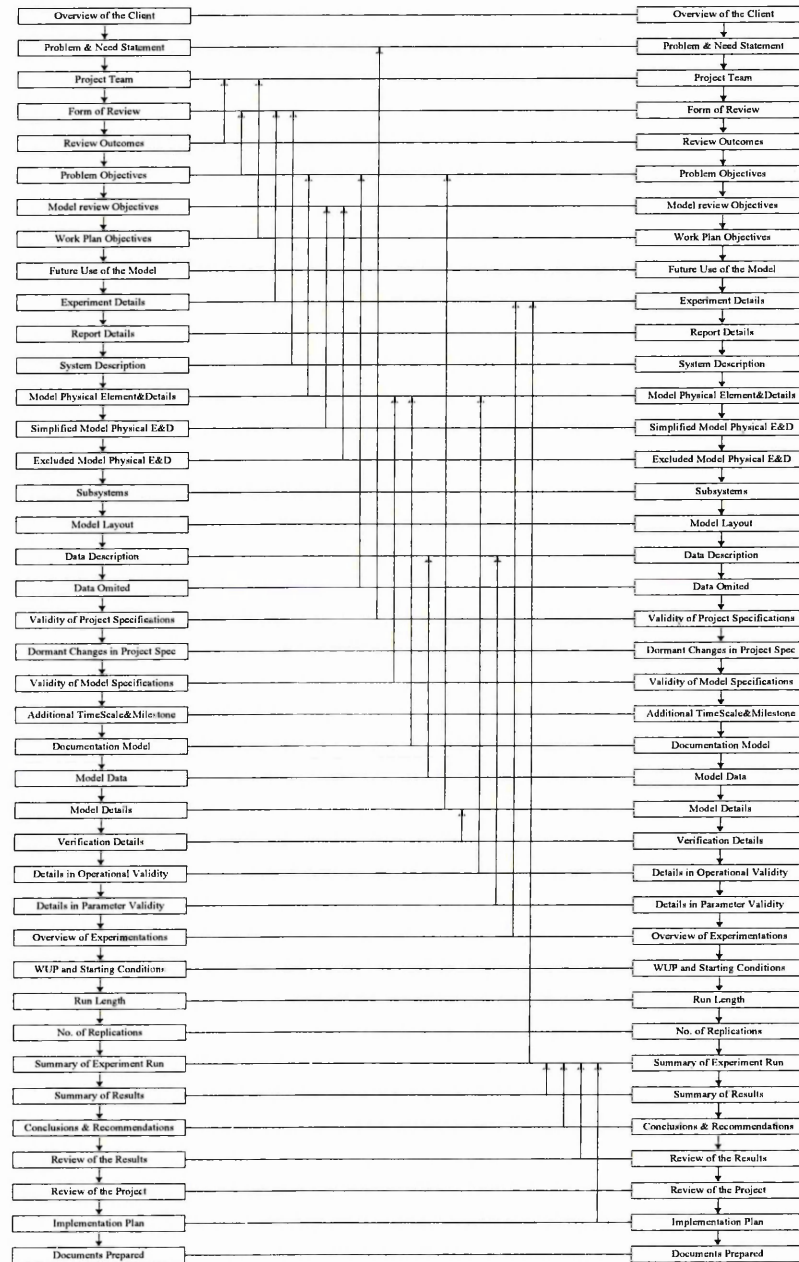


Fig. 6.14 : Alterations to Previous Task Documents in the Highest Level

6.5.2 WORD-VISIO Interface

If any system uses the same data in different levels (at least in lower levels) of the project hierarchy, such a system would be more effective. In the proposed MRD system, the model layout details, which are recorded in Word text file in the project specification subsystem, are presented to create the documentation model in Visio environment. The documentation

model, which represents the model graphically and captures model data, is then embedded into Word text file in model building and testing subsystem for recording purpose. The data in two specified locations are not in the same format. However, such hierarchical communication offers the MRD system more effective and allows maintaining the uniformity of data throughout the documentation process. Such communication is attained through the interface developed with VB as for the purpose illustrated in Fig. 6.15.

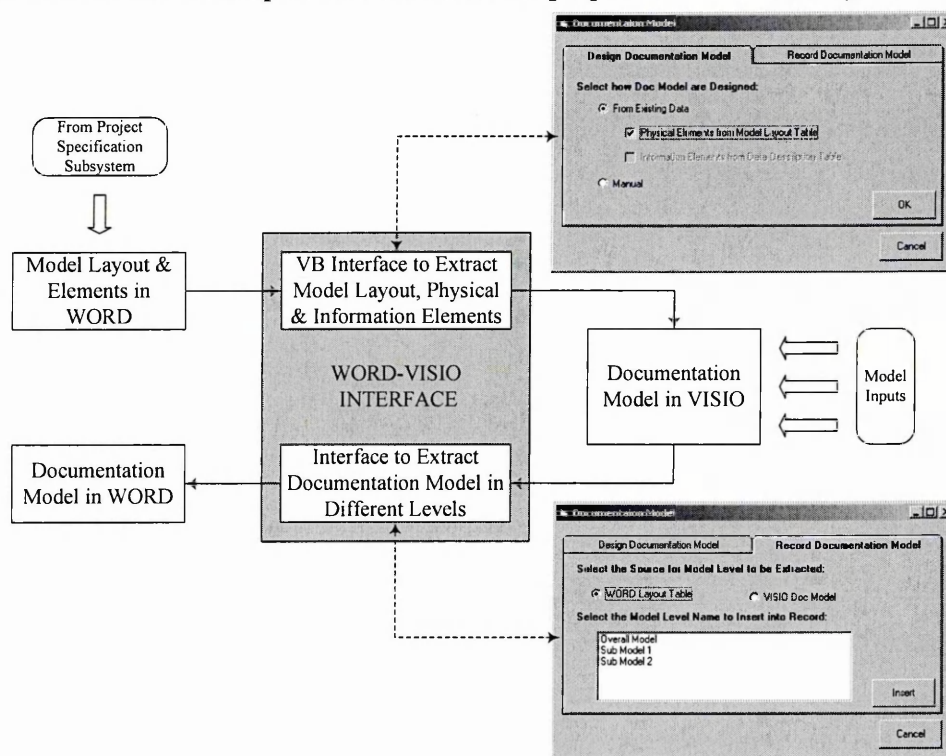


Fig. 6.15 : VISIO-WORD Interface for Hierarchical Communication

The layout details that are used to describe the interface are only one instance in hierarchical communication. Other potential details can be found in data description table (Form 5.18), by which not only the physical elements but also primary information and control elements could be presented to the documentation model. Such data sharing can be accommodated through information check box in the design display shown in Fig. 6.15. However, further development to the proposed tool is required to accommodate such data sharing. Visio to Word interface is not an essential component in documentation. Nevertheless, if such an interface allows the user to choose the different levels of the model either from layout data table or from the documentation model, whichever the convenient, then it indirectly verify the integrity of task details.

6.5.3 VISIO-ACCESS Interface

In the proposed MRD system, Access is primarily used to store the model data and the model description in a common database with standard internal structure. DM template in Visio provides the user a graphical environment to construct the documentation model, while the location of the elements, the data, and custom properties are being captured to

form the common DM database. However, the data like process times, resource schedules, resource failures, attributes, variables, etc are not the custom properties that produce the common DM database. They are stored in different databases with different formats. Visio - Access interface associated with appropriate elements allows capturing data according to the format required for documentation and model translation. All details collectively not only provide the documentation model and model data for documentation purpose, but also create a sound foundation for exchanging the model with simulation software (e.g. Arena). With this view, Visio-Access interface is required for the following tasks:

- To capture and assign model data, attributes, variables and expressions with prospective user interaction while the documentation model is being constructed.
- To export the descriptions of the documentation model and data for recording the model and data, for presenting a list of elements for verification, and for delivering details to the implementation model database with an appropriate format.

The function of the interface is illustrated in Fig. 6.16. As appeared in the left screen view (table) in the figure, the user is directed to complete data (e.g. schedule & failure data with units) with the format proposed in the data table. Some data (e.g. expression for schedule name) in the table are automatically generated, interacting with the DM database. The DM database is created by Visio built-in interface, linked through ODBC data source. As appeared in the right screen view in the figure, the attributes (e.g. process times) and variables are assigned to the model in appropriate points that are usually arrival stage of entities (e.g. Part A) or change in entity stage (e.g. Rework part A). The figure also shows the usefulness of data in DM database for model building and to the implementation model.

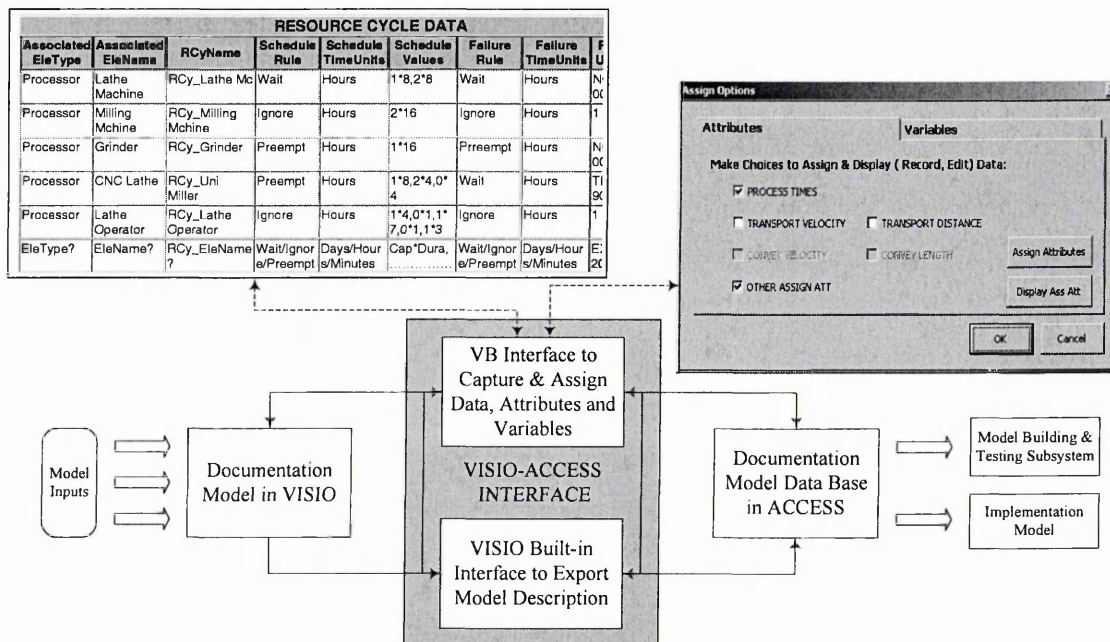


Fig. 6.16 : VISIO-ACCESS Interface for Capturing Model Description and Data

6.5.4 WORD-ACCESS Interaction

Unlike other interfaces, an interface between Access and Word is not an essential. Nevertheless, generating model data and list of elements required for model verification from Access DM database is necessary for presenting them into task documents in Word. Generation of model data tables and list of elements for verification are accomplished through SQL queries. SQL can produce a complete set of existing model data and list of elements. However, the user should be allowed to make the choice on what details that is to be documented and what elements that is to be listed for verification, depending on the nature of the model. The user interfaces developed in this respect and the interaction between Word and Access is illustrated Fig. 6.17.

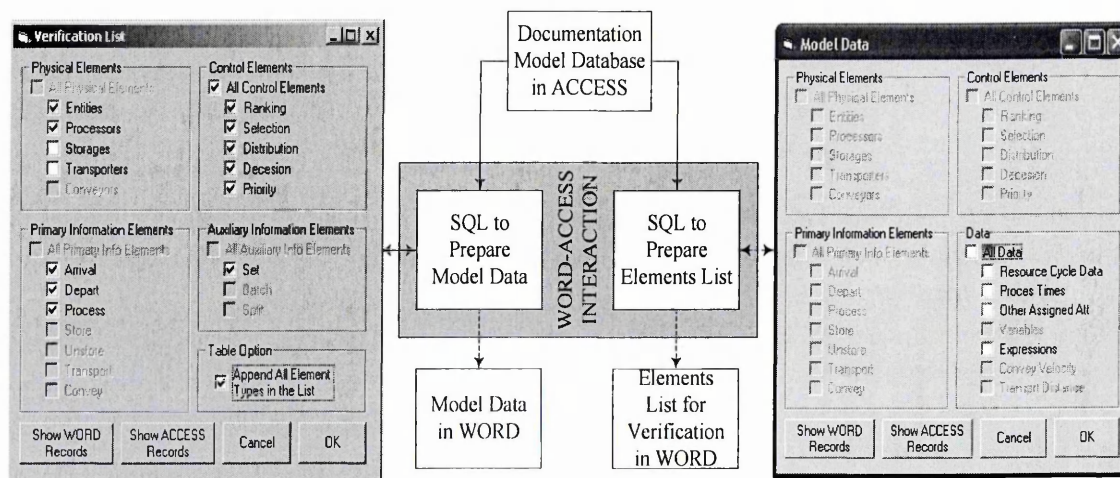


Fig. 6.17 : WORD-ACCESS Interaction for Task Documentation

In respect of model data, such a choice allows the user to provide additional information for better understanding of the model. For example, providing process details may give the opportunity for the document user to understand the notations or expressions employed in process time, clearly. However, the expressions adopted for process times in this methodology are self-explanatory (refer Fig. 6.9 and Table 6.1).

Preparation of elements list for verification from DM or DM database is significant as it makes a common structure for the verification, independently from the simulation software being used. Moreover, the documentation model is based on the separation concept or reusable model elements. Therefore, it emphasizes every component in the model, unlike in most procedural based simulation software, where elements perform more than one function. For example, as the priorities are embedded in Arena process module, the significance of this rule has disappeared, in contrast to the documentation model, in which it is a separate element. The choice given for preparing the list of elements for the verification also allows the user to accumulate them with their significance. An example presented in Annexure C illustrates the verification list generated from the DM database.

6.5.5 User Interface for Documents Preparation

This interface primarily offers the user with an environment to produce the documents for different audiences for different purposes, the phase-end documents at the end of each phase, or on selected task documents at any stage of the project. Fig. 6.18 shows the screen of the interface provided for the user to handle the documents preparation task.

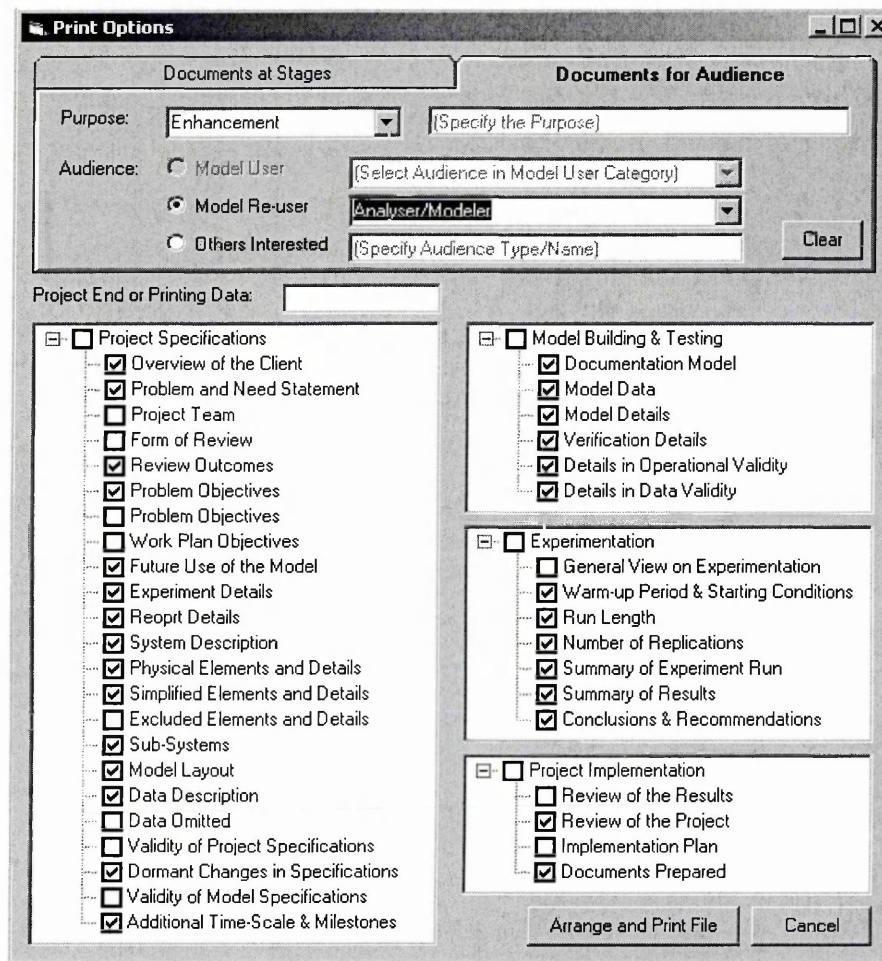


Fig. 6.18 : User Interface for Documents Preparation

The needs for different audiences and different purposes were analysed in Section 4.7.2.2 (Tables 4.5, 4.6 & 4.7). The analysis shows that the needs vary not only with different audiences but also with different purposes. This indicates that the user should be given a choice to select task documents for both purposes and audiences in preparation of documents as displayed in Fig. 6.18. With the choice made for particular type of purpose and audience, the interface displays the suggested task documents appropriately as checked items in the tree view. However, due to the variation of significance of task documents from project to project, the user should be given a flexibility to alter (add or remove task documents) final documents appropriately. In addition to the audiences and purposes stated, the user is given a flexibility to prepare documents for his or her own choice with addition of 'other' option. Preparation of phase-end is straightforward by picking right choice. Any choice made in the screen should also be displayed in the final document.

6.6 Model Exchange

Throughout the discussion made about the proposed methodology, it was highlighted that the documentation should be performed simultaneously with the model development process and both processes should directly be benefited from each other. In other words, documentation process should be integrated with model development process to make the documentation popular. The discussion also emphasized that the communication (especially hierarchical communication) makes the documentation more efficient. For example, the use of elements and data captured during the model representation for recording the model data and creating the elements list for verification, makes the documentation process productive.

Furthermore, if the output from model representation – documentation model – is translated into the simulation software, then the same model with or without a little modification could be used as the executable model. Hence, the documentation process would be more fertile and the time spent on documentation would not be a waste of time. From documentation point of view, such model translation is not mandatory. Nevertheless, it was attempted to represent the model or create the documentation model independently from the simulation software. The separation concepts and elements reuse as discussed in Sections 4.5.1 and 5.3.1 are the major drive behind software-independent model representation. Once the documentation model is created independently, it is then translated into a model in procedural based or object oriented simulation software through relevant interfaces. In this respect, the common DM database discussed previously acts as media for model translation.

Model translation process is referred as ‘pre-documentation’ as the initiation to create the executable model commences with documentation model. Once the model is translated, it is not strange if any alterations (removal or addition of elements) to the model take place at any time in any stage. Such situation raises a need for updating the documentation model. Updating process is referred as ‘post-documentation’. The basic concept of model exchange is diagrammatically shown in Fig. 6.19.

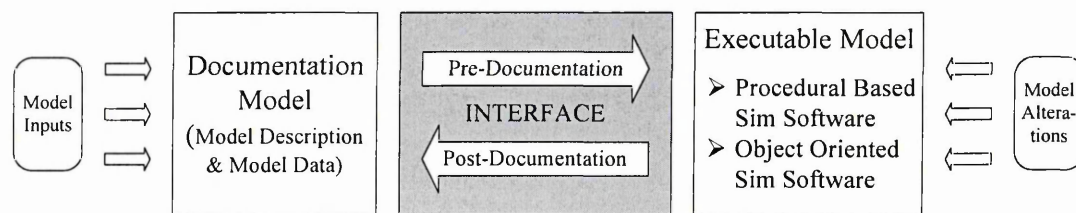


Fig. 6.19 : Exchange of Documentation Model with Simulation Software

As illustrated in the above figure, model exchange function is performed with appropriate interfaces between the documentation model and the simulation software that is intended to use. Interfaces may be different from one to another as the individual software offers varying structure for model construction and interfaces with other software packages such

as Access, Excel to manipulate the model and the model data effectively. The feasibility of the model exchange process in the proposed methodology has been studied with Arena. Arena has been selected for this purpose due to the following reasons:

- Arena is the most commonly used simulation software package in both academic and industrial field (Al-Ahamari and Ridgway, 1999; Kovács, et al., 1999; Perera and Liyanage; 2000; Anglani, et al., 2002).
- It is one of the most cited software in the literature in respect of the research work like development of generic pattern in reusable form, design of object oriented interfaces, etc. (Eldabi and Paul, 1997; Arons and Asperen, 2000; Anglani, et al., 2002).
- Its special features are more useful for modelling manufacturing system. However, it can be applied for other applications such as health care system (Tesham and Unger, 1996; Manansang and Heim, 1996), business process modelling systems, etc.
- Arena is classified as a layered system or hybrid system that provides more flexibility for the user to build the model with building block as well as programming. Therefore, it displays the features of simulation languages as well as VIM software packages.
- As mentioned by Takus and Profozich (1997), Arena is based on OO paradigm. However, its simulation engine is SIMAN, which is generally accepted as procedural based. Accordingly and with the features displayed by Arena, it seems that Arena is ‘partially object-oriented’ and displays features of both procedural and OO paradigm.

6.6.1 Model Exchange with ARENA

Arena offers sound interfaces such as ODBC, OLE, VBA, and DXF to accelerate the modelling with more convenience environment. It has a built-in option to export a model to a database or import a model from a database (Access/Excel) via ActiveX dynamic libraries. Regardless of the database utilized, this option allows the model data to be organized in a standard set of tables. The Use of this standard set of tables with built-in “smImportFromDatabase” dynamic link libraries become the drive for translating the DM into Arena - pre-documentation – in the proposed MRD system. Whereas, incremental VBA interface in Arena is used to update the documentation model for post-documentation purpose. Fig. 6.20 exemplifies the overview of the model exchange process with Arena and how and what task documents associate with the model exchange process.

Although, task recording appeared in the figure, it has already been dealt in-detailed previously. Therefore, model exchange process is now explored in-depth in order to illustrate how the documentation model is translated into the implementation or executable model and how the alterations made in the executable model are captured for updating the documentation model. It is clear from Fig. 6.20; Access databases act as the media for model translation as well as for recording some tasks.

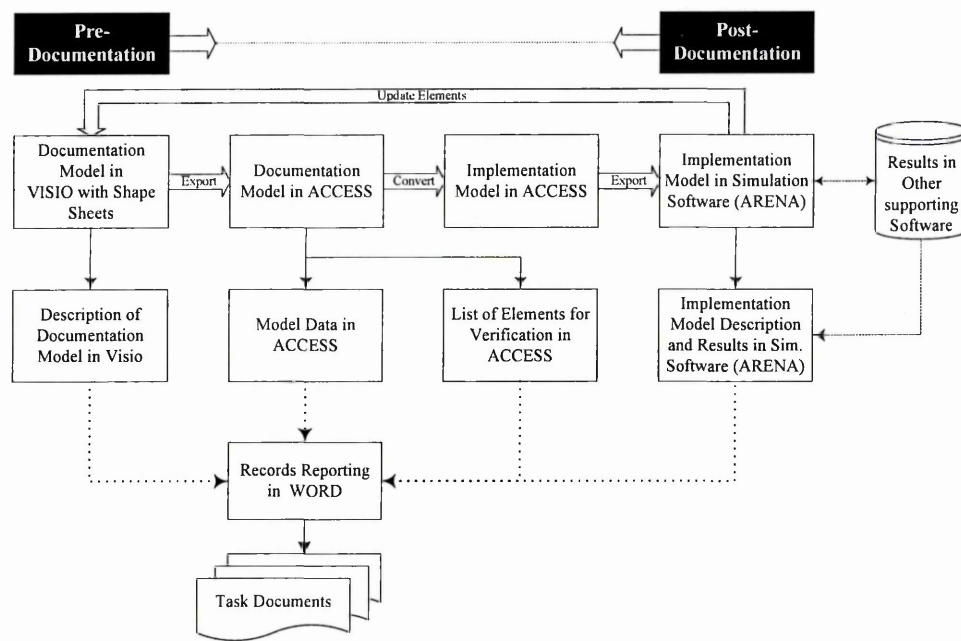


Fig. 6.20 : Overview of Model Exchange and Task Recording

Translating (pre-documentation) and upgrading (post-documentation) the DM are accomplished through built in interfaces in Visio and Arena, and VB interfaces that were developed for specific purposes. Exporting DM into its database and implementation model database into executable model is performed with the use of built in facilities in Visio and Arena respectively. However, due to lack of cohesion between the structure of the DM database and the standard Arena implementation model database, VB and VBA interfaces were developed to convert DM description and data into standard set of database tables, expected by Arena, before translating the DM. Moreover, VBA interface was built for updating the DM at minimum level (i.e. elements with names only) with an assumption that a fewer alterations is expected due to proposed extensive approach to model building and documentation process. On the other hand, an extensive update may not be possible or even much complicated. Complexity of such interfaces depends on how the different types of elements in the software are structured and what external internal interfaces are provided in the particular software. With this view, interfaces for pre and post-documentation that are developed for exchanging model with Arena are briefly explained in Fig. 6.21.

6.6.1.1 Functions and Limitation in Pre and Post Documentation with ARENA

The functions and limitations of pre and post-documentation with Arena are explained with an example, presented in Fig. 6.22. It illustrates the output from pre and post-documentation for a system that corresponds to process related activities. As appeared in the figure, the DM (left) created in Visio together with data captured with Access are translated to Arena model without connections and built-in variables. Translated model could then be executed directly with few alterations. The same approach could be extended for material handling related models with further development to the DM template, DM stencil, and interface.

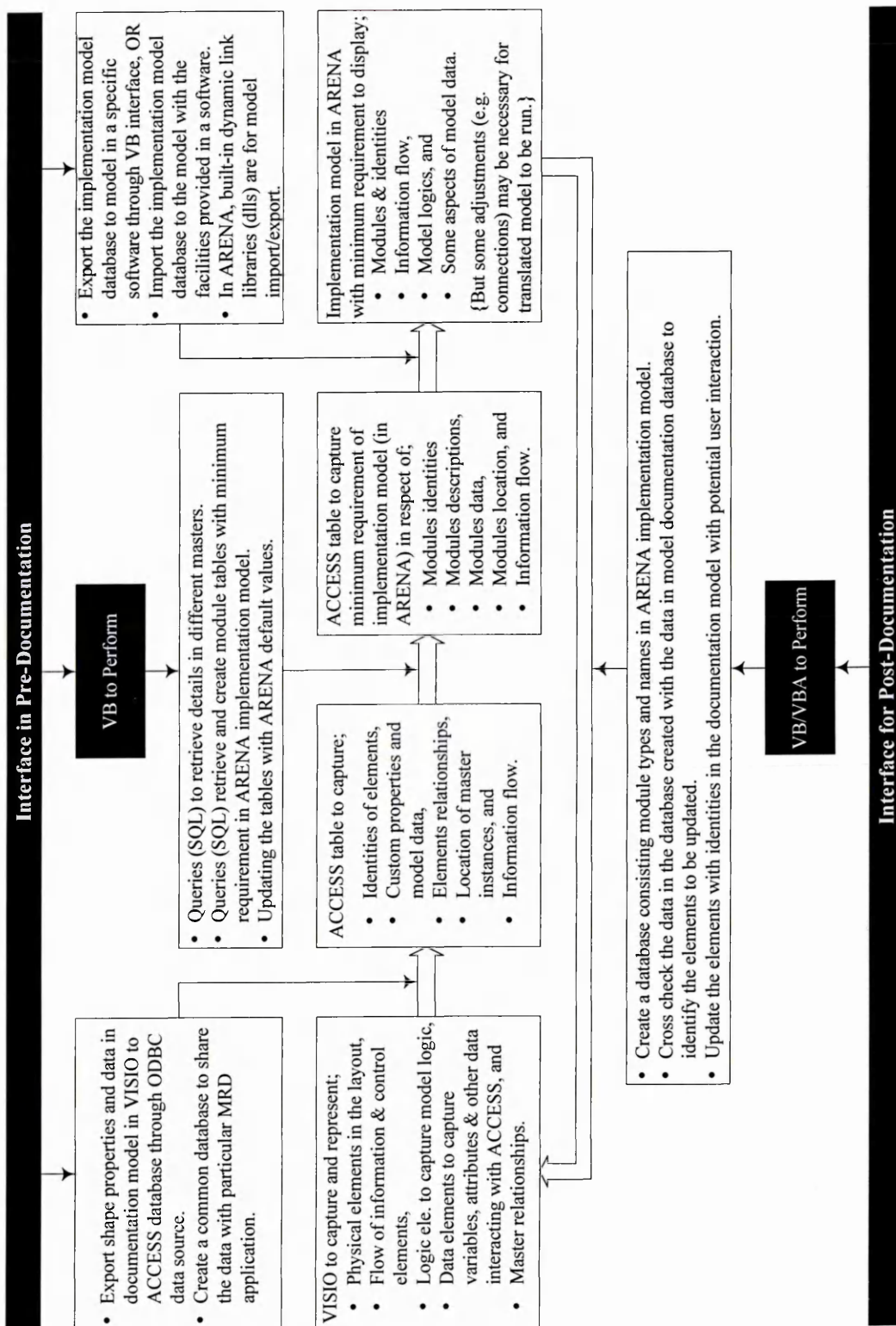


Fig. 6.21 : Detailed View in Model Exchange with ARENA

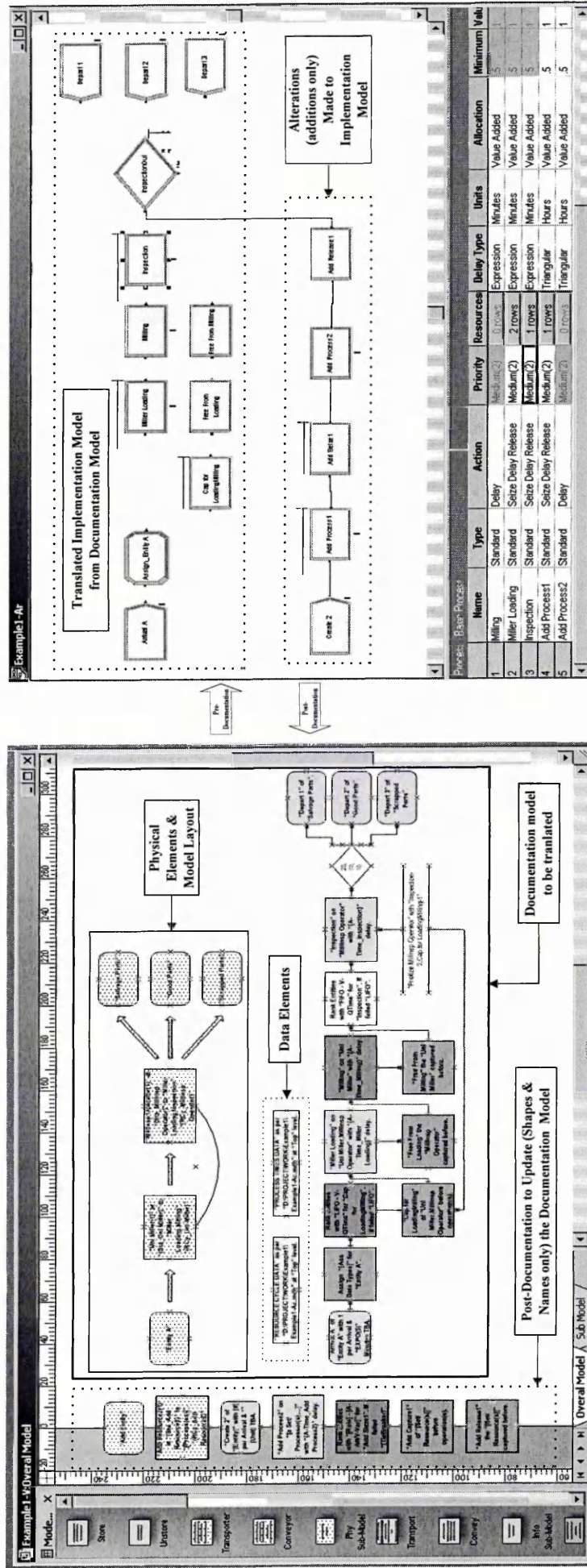


Fig. 6.22 : Functions and Limitation in Pre- and Post-Documentation with ARENA

As discussed before, alterations to the implementation model could be expected at any stage of the project. Therefore, the translated model has been altered by adding more modules (assume that deletion of modules is not encountered in this example). The documentation model is then updated with minimum requirements (shapes and names only). During the updating, the shapes are located along the Y-axis, overlapping the same types of elements together. The updated elements can then be re-located appropriately. However, pertinent modifications to the element data sheets or data elements should still be taken separately. In case of any removal of modules from the translated model, appropriate elements in the documentation model are deleted with user interaction. Nevertheless, relevant data (e.g. process time) may yet to be removed individually and manually.

6.6.1.2 Mapping Tables

It is appeared in Figs. 6.20 and 6.21 that Access database or tables are vital components in model exchange process. More specifically, it is evident from Fig. 6.21 that VB interface for pre-documentation is primarily associated with queries (SQL) to extort the data from the DM database and present them into a standard set of tables in Arena implementation model database. In generic terms, database tables lay a foundation for exchanging the model with the simulation software. This highlights the significance of introducing mapping database tables to integrate the DM and the implementation models within a common framework.

The DM database comprises of custom proprieties of the elements for which some data may have been assigned. Arena implementation model database encompasses with operands and data of the modules. However, there are structural differences between the elements in DM and the modules in Arena. Hence, one-to-one mapping becomes impossible in translating the model. Therefore, a necessity of an appropriate interface for automating the model translation would arise. The best way of initiating the interfaces for model translation with Arena is by introducing mapping tables to map constitutes of both models. The similar conceptual approach may be applied for translating the DM into other simulation software.

Mapping the details of elements in DM with modules in Arena model may not replicate the same process as with other software. However, the details mapped with Arena are described for the illustration purpose of the concept. Nevertheless, as the details to be mapped produce an exhaustive list, only an example is presented for illustration purpose. The complete list is presented in Annexure A. Table 6.3 shows the mapping table for Arena 'BasicProcess|Process' standard table that directly represent 'process' module in Arena. It is assumed that the "process" module is performed "Delay" and "Seize, Delay, and Release" actions only. Table 6.4 describes the details of 'BasicProcess|Process|Resource' table that illustrates how other DM elements correlate with Arena 'process' module.

ARENA Executable Model Database					Documentation Model (DM) Database				
Table Name	Field Description	Field Name	Prompt	Prompt	Field Name	Field Description	Element	Method of Capturing Values	
Basic Process Process	Unique ID for a 'process' module	SerialNumber	Auto	Auto	ShapeKey	Unique ID for an instance of a master	PIE -Process (/LE-Delay)	When the element is exported to database	
	ID for model level	ModelLevelID	Auto	Auto	P_ModelLevelID	ID for model level		Element property, extracted from layout model level or page property	
	X – Module location for 'process' module	X	Auto	Auto	PinX	X – Element location		When the element is exported to database	
	Y – Module location for 'process' module	Y	Auto	Auto	PinY	Y – Element location		When the element is exported to database	
	Unique name for process in 'process' module	Name	Input	Input	P_ShapeName	Unique name for process		Element property that user is expected to input	
	Type (only D/SDR) in 'process' module	Action	Select	Select	P_OperationCategory (= D or CDF)	Category of the process		Property - Choice with right click and differentiate with colour (def=Process)	
	Method of specifying parameters in 'process' module	DelayType	Select	Select	P_TimeBehaviour (=Expression)	Process time by expression		Element property – Fixed value is 'expression'	
	Name for time expression in 'process' module	Expression	Input	Auto	P_ProcessDelayID (Time_P.ShapeName)	ID for process time or delay		Element property – Auto generated with adding 'Time_' to shape name	
	Time units for delay in 'process' module	Units	Select	Input	P_TimeUnits	Format of process time units	DE-Process Times Data	Input value to the table produced interacting with the element.	
	Priority of entities waiting for a same resource in 'pro'	Priority	Select	Select	P_OerationPriority	Priority when more than one process	CE - Priority	Element property that the user is expected to input with specific function	
	Method of specifying in 'process' module	Type	Select				-- "Process" element itself is a standard element --		
	Collect statistics or not automatically in 'process'	ReportStatistics	Checked/Unche.				-- Ignored --		
	Allocation of process time or cost in 'process'	ValueAdded	Select	Select			-- Ignored --		

Table 6.3 : Mapping Table for ARENA BasicProcess|Process Standard Table

ARENA Executable Model Database				Documentation Model (DM) Database					
Table Name	Field Description	Field Name	Prompt	Prompt	Field Name	Field Description	Element	Method of Capturing Values	
Basic Process Process Resource	Unique ID for a 'process' module	ModuleSerial Number	Auto	Auto	ShapeKey	Unique ID for an instance of a master Type: Processor or Set	PIE -Process (/LE-Delay)	When the element is exported to database and then VB for indexing	
	Type (=Resource/Set) in 'process' module	Resource Type	Select	Select	P_IndexType			Element property that user is expected to select	
	Unique name for resource set in 'process' module	Set Name	Input	Input	P_IndexNames	Unique name for a set selected		Element property that user is expected to input , if Type is 'Set'	
	Names of the resource(s) input in 'process' module	Resource Name	Input	Input	P_IndexNames	Unique name(s) for processor(s)/in a set		Element property that user is expected to input, if Type is 'Processor'	
	Names of the resource(s) input in 'set' module	Set Name	Input	Input	P_IndexNames	Unique name(s) for processor(s)/in a set	PIE-Set	Element property that user is expected to input as members of the set	
	Index No. when more than one resource	Resources Index	Auto	SAuto	P_IndexNos	Index No. for processors (def=1)		"1" as element property; but indexing for each processors	
	Index No. for each member of a resource set		Auto	SAuto	P_IndexNos	Index No. for processors (def=1)	PIE -Process (/LE-Delay)	"1" as element property; but indexing for each processors in the set	
	Selecting method of a member of a set in "pro" M	Selection Rule	Select	Select/ Input	P_Rule	Selecting method of a member of a set	CE - Selection	Element property that user is expected to select or input with variable/attribute	
	Number of resources/set in 'process' module	Quantity	Input	Input	P_NoOfEntitiesPer UnitRes	No. of entities processed in a unit	PE - Processor	Element property (def=1); user is expected to input (under development)	
	Attribute name to save index numbers	Save Attributes	Input	Input			-- Ignored --		
	Input for attributes	Input Attribute	Input	Input			-- Ignored --		

Table 6.4 : Mapping Table for ARENA BasicProcess|Process|Resources Standard Table

Notes for Tables 6.2 & 6.3:

PE	Physical Element	CE	Control Element	DE	Data Element
PIE	Primary Information Element	AIE	Auxiliary Information Element	P_	Property in Visio Data Sheet

It is materialized from Table 6.3 that Arena process module is mainly formed by the 'process' (a Primary Information Element – PIE) DM element. It acts either as a process (capture, delay, free) or delay for building the model logic. However, process times, which are assigned as attributes to the entity, are captured from the 'process times' (a Data Element - DE) DM element. It produces process-time data table for the entire model for recording purpose. In attempting two different entities or two different stages of a same entity to capture a processor (or a resource in Arena) for different operations, priorities for different operations are captured from the 'priority' (a Control Element – CE) DM element.

The significance of Table 6.4 is that the selection rules are extracted from 'Selection' (a Control Element – CE) DM element. Selection element has been separated to emphasize the operation selection criteria for recording purpose. Overall, it is clear that one-to-one mapping is not possible in the proposed approach, though there are similarities at a glance. Such detailed mapping directs the development of appropriate interfaces for automating the model translation. Hence, making the DM and MRD process is more resourceful.

6.6.2 Model Exchange with Other Simulation Software

The different software may have different structures and provide different user interface for constructing the model. They may have facilitated with various external interfaces to accelerate or automate the modelling process. However, all of them possibly endows with model databases to manipulate the model details that can be extracted from the proposed the common DM database. Therefore, with sound development of appropriate interfaces, the DM could be able to translate into software-dependent executable model, following the approach similar to Arena. This context also indicates interoperability of simulation models.

6.8 Summary

This chapter has primarily presented the proposed framework for documentation of simulation projects in parallel with the simulation model development that directly benefits from documentation. Having identified the different types of details and the needs for functional and hierarchical communication for an integrated documentation system, it was discussed, what software are incorporated for the MRD system and how potential subsystems communicate for accelerating the documentation. In attempting to elaborate the communication between subsystems and other software, it was revealed that the DM is a core element in documentation. Hence, the DM was discussed explicitly in respect of its software-independent development and exchange with Arena. Due to different nature of software handling to capture task details, and the iterative nature of simulation project tasks, the MRD tool developed for on-line documentation was discussed. The functions of interfaces and features embedded to or expected from the MRD tool were also emphasized.

7. VALIDATION OF THE PROPOSED METHODOLOGY

7.1 Introduction

As discussed in Chapter 03, conclusions can only be made after testing the proposed methodology on different real-world situations and then making the appropriate alterations in order to present a more realistic solution to the documentation in simulation. With this view, action research can be considered as the most appropriate strategy to validate the proposed methodology. However, such a rigorous validation is beyond the scope and is left for further research, as it requires applications of the methodology for different scenarios in diverse disciplines. In addition, such a validation is hampered due to the limitation in the MRD tool developed, particularly in the construction of documentation models for different problem domains and the model exchange function.

Therefore, as discussed in Sections 3.3.2.3 and 3.3.2.4, a structured questionnaire survey, which is supported by a case study as an example to demonstrate the concept, is considered as an alternative to the action research. The questionnaire survey primarily aims to obtain the views and opinions from experts on; how effective the proposed methodology would be in real-world situations, and what further improvements are necessary. The methodology is still in its development stage, and therefore, experts' views from both academia and industry are worthwhile to reflect the findings from both standpoints.

Therefore, this chapter begins with a brief description about the specific objectives, types, and format of the questionnaire and the boundaries of the case study. The results of the questionnaire are analysed, emphasizing the key points of the proposed methodology. The positive and negative aspects of the proposed methodology that are identified from the results are then discussed in respect of implementation and further improvement.

7.2 Questionnaire

The questionnaire (Annexure B) consisted of following seven major components. They aimed to obtain the details of the respondents and their opinion about the development and implementation aspects of the proposed documentation methodology.

- Descriptions of the participants,
- Current practice and general opinion about documentation in simulation,
- MRD process and proposed approach,
- Proposed MRD system,
- Integrated MRD system with model development,
- Overall assessment on the methodology, and
- Other remarks regarding the proposed methodology and documentation in simulation.

The questionnaire consists of highly technical materials about the proposed methodology. Therefore, great care was taken to present the questions as self-explanatory. New terms and the concept were explained through ‘supplementary details’ that were provided at the end of the questionnaire and they were hyper-linked in appropriate questions. The methodology was further described through an example that demonstrates the application of the methodology to a case study. It was intended to conduct structured interviews to enhance the methodology through obtaining additional feedback to the questionnaire, but this was not possible to put into practice completely due to busy participants. However, telephone conversations and email messages have made the survey successful to a great extent.

7.2.1 Specific Objectives of the Questionnaire

In Section 3.4, documentation of simulation projects was defined and the characteristics of a documentation system were established. Based on the new definition and the characteristics, a methodology to document simulation projects was proposed. However, those characteristics are based on both the literature in simulation and other related areas, and the knowledge gained through the learning process. Hence, they can be considered as assumptions to be validated. In this respect, the objectives of the questionnaire are:

- to determine the current practices in documentation of simulation,
- to ensure that the definition for documentation of simulation project is appropriate,
- to certify that the characteristics incorporated for documentation process are valid.

As suggested in Section 3.4.2.2, the proposed approach to record the details is ‘task-orientation’. Task documentation should be based on the simulation project that may not be conducted with the same practical approach by two different individuals. However, in order to propose a common structure for documentation, it was assumed that the conceptual approach to any simulation project remains the same, and therefore, a revised project procedure was proposed for identifying the task documents. Accordingly, the task documents were identified, and the structure and format for maintaining uniform and consistent details were presented in Chapters 04 and 05. Therefore, the next set of the objectives of the questionnaire are:

- to confirm the appropriateness of the revised or standard project procedure for identifying task documents,
- to substantiate the appropriateness of the task-oriented approach to document the details of simulation projects,
- to evaluate the advantages and disadvantages of task orientation for documentation.

It was assumed that the documentation should be independent from the simulation software for facilitating different categorizes and levels of audiences. Software-independent task

details were addressed with the use of 'reusable model elements' that collectively represent the behaviour and the logic of the model graphically and capture model data. In respect of software-independent documentation, the objective of the questionnaire is:

- to ensure that the reusable model elements describe the model explicitly and represent the model (documentation model) independently from the simulation software.

Having explored the task documents and their relationships, a documentation system (referred to as MRD system), which has a common layout and systematically linked and interfaced details in each task of the project, was proposed. The proposed MRD system is expected to help and guide the project team to record details as the simulation project progresses. Hence, the next objective of the questionnaire is:

- to certify that the facilities, which have been incorporated in the MRD system, help the modellers to document simulation projects.

The MRD system was interfaced with the model development process so that the executable model could be generated from the documentation. Such an integrated MRD system is anticipated to improve the documentation further, enabling the modellers to treat the documentation as an integral activity of the simulation project. Therefore, subsequent objectives of the questionnaire are:

- to confirm that the integrated MRD system improves documentation in simulation,
- to substantiate that the proposed methodology is generic and standardized,
- to validate that the proposed methodology is practical and viable.

In addition to the benefits in respect of the documentation in simulation, the proposed methodology may also offer additional benefits to the simulation community. However, the methodology revisits weakness that may have not seen or directly visible within the limited knowledge and experience possessed. Therefore, this questionnaire is also intended:

- to endorse the additional benefits expected from the proposed methodology, and
- to explore the weaknesses of the proposed methodology.

On achieving the above objectives, the methodology could be validated and further improvements to the methodology could be suggested.

7.2.2 Format of the Questionnaire

The questionnaire consists of 25 questions of various types, including multiple-choice, short answers, checklists, ratings, and free-form questions. Multiple-choice and checklist types of questions primarily aim to ascertain the facts in respect of the current practice in documentation. Rating and free form types of questions aimed to determine the effectiveness of the proposed methodology for documentation of simulation projects, and to

establish the characteristics and relationships between the proposed approach and current practice. Apart from the specific questions, the participants were invited to express any positive or negative views about the proposed methodology and any other comments that they consider useful in respect of documentation of simulation projects.

7.3 Case Study

Questionnaire survey was supported by a case study as an example to illustrate the application of the documentation methodology to a real-life situation. However, due to the limitation of the MRD tool developed, the example did not represent an actual case study. Rather, a hypothetical case study was designed to suit the existing capabilities of the MRD tool. Therefore, it did not attempt to present the technical contents of a real problem. Instead, it portrayed how the proposed method is performed in parallel with the model development process in order to generate a complete document of the simulation project. Several assumptions were made in the example to suit the capabilities of the MRD tool, though they may not be realistic in a real-world environment. However, from the complete document, presented in Annexure C, it is anticipated that the reader can understand; how the simulation process was performed, and what results are obtained in each task, without any further explanation, unless there is no misinterpretation of the technical contents.

7.4 Participants

The questionnaire consisted of both general and specific questions with highly technical contents. Therefore, it was anticipated to conduct the survey with structured interviews or to provide further clarification over the telephone and/or email messages. Consequently, the participants were selected from known experts to the research team. Hence, the survey was limited to a small number (12) of participants. As the methodology is still in its development stage, both academia and industrial (internal and external) experts were solicited to participate in the validation process. According to the survey results, most of the participants have built more than five simulation models, and 83% of them have knowledge and experience of using Arena simulation software. A different individual has experience with other software like Witness, ProModel, Quest, Simula8, etc. as well.

7.5 Survey Results and Analysis

The results of the questionnaire are presented in Annexure D. The results are then analysed and concluded to show how effective the proposed methodology would be in real-world situations, according to the knowledge and experience of the respondents. The analysis is taken place in line with the sequential development of the proposed methodology. Although, question 01 was used to obtain the participants personal details and is not used as part of this analysis.

7.5.1 Current Practice and General Opinion

This section highlights the current practices in documentation of simulation projects and the widely held opinion on documentation in simulation.

Question 2.1 asked whether the participants generally document the simulation project. No respondents answered 'No' to the question, but as shown in Table 7.1, it seems that documentation in simulation is rarely considered as a compulsory activity. However, the project details are generally recorded by some means.

Response	Percentage
Always 'yes'	08%
Usually 'yes'	42%
Sometimes 'yes'	42%
Seldom 'yes'	08%
Always 'No'	00%

Table 7.1 : Are the Project Details Documented?

Question 2.2(a) enquired the level of details generally recorded. The results show that 92% of participants record moderate project details to assist the project team for the modelling process, experimentation, and model implementation. The rest of the participants records comprehensive details for the benefit of all team members, model re-users and other interested parties. These results clearly show that modellers do not completely ignore the documentation and record with varying degrees of details due to various reasons.

Question 2.2(c) attempted to discover the reasons or barriers for not documenting project details or varying effort in documentation. Table 7.2 summarizes the reasons and the responses in percentages.

Barriers for Documentation	Percentage
Non-availability of structured methodology and guidelines	75%
Lack of time for proper and detailed documentation	58%
Isolated documentation process from the model development	42%
Difficulty in identification of significant details	25%
Tendency to forget details in case of retrospective documentation	17%

Table 7.2 : Barriers for Documentation of Simulation Projects

As in Table 7.2, non-availability of standard and structured method for documentation is the foremost barrier for documentation in simulation. In addition, 92% of participants, who responded for question 2.4(c), strongly agreed (17%) or agreed (75%) that the documentation system should have a common layout with structured and linked details. This evidence was the major motivation behind in this study, and was comprehensively addressed in the proposed methodology. The second reason is the lack of time for proper and detailed documentation. This is an outside-issue that cannot be addressed directly by the methodology. Nevertheless, an attempt to encounter the third reason (i.e. isolated

documentation) indirectly reduces the time for documentation. Therefore, the documentation process was interfaced with the model development process in order to overcome the second and third obstacles in documentation. A need for such an integrated system was substantiated by all participants (67% - strongly agreed and 33% - somewhat agreed), who responded for question 2.4(d). Although, the last two reasons appear to be less significant, they have been clearly addressed through progressive task documentation in the proposed methodology.

The results of the question 2.2(b), regarding the stage of the documentation, depicts that 17% of respondents document the details progressively and 08% record details retrospectively. Nevertheless, the majority (75%) attend to the documentation progressively and retrospectively, depending on the details. Therefore, it seems that progressive and retrospective documentation is the usual practice. Nevertheless, according to the results for question 2.4, 92% of respondents strongly agreed (50%) or agreed (42%) that the documentation should be progressive, and thereby the assumption regarding the progressive documentation can be considered as valid.

Questions 2.2(b) and 2.4(b) were to ascertain the effect of the software on project details. Most of the participants had not responded to the question 2.2(b) that may be due to lack of details presented in the question. Therefore, the effect of the software on current documentation practice could not be ascertained. According to the responses for question 2.4(b), participants had mixed level of agreement about the software-independent details. Overall, they neither agreed nor disagreed that the documentation should be independent from the simulation software. Undoubtedly, software-independent details have a great impact on the development of a generic and structured documentation process, which cannot be implemented with software-dependent details. In the proposed methodology, software-independent task details were addressed with the use of 'reusable model elements'. Nevertheless, a need for software independent details should be comprehended with further discussion and research.

7.5.2 How Effective the MRD Process?

This section describes; how appropriate the new definition for documentation in simulation, how useful the task-orientation for documentation simulation projects, how successful the reusable model elements for software-independent documentation, how significant the pre-structured forms for presentation of task details, and how viable the proposed approach.

Due to varying interpretation and perspectives on documentation in simulation, documentation was defined to fulfil the various needs of different audiences. Questions 3.1 aimed to substantiate how appropriate and how useful the new definition in a real-world

situation. Although, the participants had mixed opinions about the new definition, overall, the new definition is accepted to a great degree. In addition, 83% of respondents agreed that the task-orientation is the most appropriate approach for documentation of simulation project for fulfilling different needs of the audiences. The rest of the participants neither agreed nor disagreed the appropriateness of task-orientation approach for documentation.

In the proposed methodology, the task documents were created by breaking the revised (standard) project procedure into micro-level of tasks, assuming that the conceptual approach to any simulation project remains the same. A great attempt was made to present the details of task documents by means of forms. 75% of the participants accepted that a standard project procedure is necessary to propose a generic documentation framework that enables the documentation to apply to any simulation project. In respect of benefits from task documents and their pre-structured forms, almost all participants agreed that:

- task documents serve as a checklist for documentation,
- pre-structured forms provides a uniform structure for recording task details,
- pre-structured forms encourage/help modellers for documentation,
- pre-structured forms guide the modellers for simulation projects.

The use of 'reusable model elements' is the basis for software-independent task details in the proposed methodology. Questions 3.6, 3.7, and 3.8 aimed to evaluate how useful such elements for documentation in simulation. Similar to the results in questions 2.2(b) and 2.4(b), respondents had positive and negative reactions about the reusable model elements and software-independent details. Nevertheless, almost all participants believed that the proposed reusable model elements help the modellers to describe the model explicitly.

7.5.3 How does the MRD System Assist?

This section confirms how the facilities that were incorporated in the MRD system assist the user to prepare documents of any simulation project with a common layout. Moreover, any other facilities that were suggested by the participants are presented so that such proposals can be accommodated in further research.

In the MRD system, the documentation templates that comprise of pre-structured forms of task documents, maintain a common or standard layout for documentation. Question 4.1 asked how helpful the documentation templates for documentation of simulation projects. All participants accepted that the documentation templates encourage/helps modellers to maintain documentation and to produce documents with a common layout. They agreed to a lesser extent that the repetitive details should be shared for an efficient documentation system. However, the detail sharing is significant to reduce of time needed for recording repetitive details, which is an obstacle in documentation.

Handling task documents is a challenging issue of the MRD system to be efficient. Therefore, the MRD system was incorporated several facilities in order to help the modellers for different purposes. The questionnaire aimed to evaluate how appropriate those facilities and to acquire any other facilities that should be incorporated to the system. Table 7.3 shows how effective those facilities for the proposed system according to the opinion of the participants. The survey substantiated that almost all the facilities that are shown in Table 7.3 are appropriate for an efficient MRD system. No other facilities were suggested in respect of task handling.

Facilities for Handling Task Documents	Percentage
Display different stages of the simulation project and task documents	67%
Flexibility to access any task document at any time/stage of project	67%
Attention to previous task documents that need alterations	67%
Visualize the contents of task documents immediately	58%
Distinguish between recorded and unrecorded task documents	58%
Alterations to task details with least interactions	33%

Table 7.3 : Facilities for Handling Task Documents

In addition to those facilities in task handling, participants were invited to express their opinion about other facilities that have already been incorporated, or to suggest any other to make the MRD system more efficient. 83% of participants recognized that the MRD system should facilitate to produce documents for different purposes and audiences at any stage of the project. Similar to question 4.1, sharing repetitive details had not been recognized (only 43% accepted) as a significant facility in the MRD system, although data sharing notionally reduces the recording time. Nevertheless, provisions for presenting the document with PowerPoint and Web have been suggested by the respondents for a useful MRD system.

7.5.4 How Effective the Integrated MRD System?

This section describes how the integrated MRD system contributes for the documentation and model development process. It also aimed to evaluate the feasibility of using a standard MRD system within different application domains and with different simulation software.

The integrated MRD system primarily facilitates to translate the software-independent documentation model to a software-dependent executable model and vice-versa through a 'common database'. According to the results of question 5.1(a), all respondents are with the opinion that the integrated MRD system encourages and helps the modellers with documentation. Similarly, all participants substantiated that the use of a structured common database makes provision for model reusability. However, due to insufficient detailed description presented in the questionnaire, the participants responded neutrally (neither accept nor reject) about the strength of the proposed common database in an integrated documentation environment.

Having studied the feasibility of the integrated MRD system with Arena software, a standard MRD system was proposed in order to generalize the approach. However, such a conclusion was based on very limited exposure to different application domains and other simulation software. Therefore, questions 5.2 and 6.2 attempted to substantiate such a conclusion from the participants that have experience in different domains and of using different structural simulation software. From the overall results, it appears that such a standard system is practical and viable. As expected, all participants who have experience with Arena agreed that such an integrated system is highly likely. Those who have experience with Witness consider it as a candidate for such a system. However, the development of individual interfaces for different software is considered as a weakness of the integrated MRD system.

7.5.5 Does the Methodology Offer Other Benefits?

This section discusses the experts' views about the other benefits from the proposed methodology. In addition to the confirmation of additional advantages, which have already been identified, any other benefits that were raised during the survey are explored in this section in order to authenticate the usefulness of the methodology.

A part of the question 6.1 was to authenticate the additional benefits from the proposed methodology. Table 7.4 illustrate to what extent the respondent agreed with additional benefits that have already been identified.

Additional Benefits	Strongly Agree	Agree	Agree/ Disagree	Dis- Agree	Strongly Disagree	Overall Results
The proposed system acts as a management, control, and monitoring system of simulation projects.	17%	83%	00%	00%	00%	Agree
The proposed system acts as a training system for a novice simulation practitioner.	16.5%	67%	16.5%	00%	00%	Agree
The documentation model with the common database enhances the model reusability.	00%	75%	25%	00%	00%	Agree

Table 7.4 : Additional Benefits from the Proposed Methodology

In addition to the benefits discussed above, participants also suggested other potential benefits that can be expected from the proposed methodology. Among many, two seem to be more significant. The proposed documentation, which has a common layout with systematically linked details and interfaced modelling process, increases the value of the simulation project and shortens duration of the project life cycle.

7.5.6 Weaknesses and Limitations of the Methodology

This section discusses the negative opinions presented regarding the proposed methodology. The weaknesses will be explored with the view of defending them within the present status of the methodology or improving the methodology later as further research.

It was a realized fact that the task documentation (or the MRD process) is lengthier and more time-consuming process than the usual practice. Overall, participants agreed (17%-strongly agreed and 50% - somewhat agreed) the fact that the proposed MRD process is lengthier and time-consuming. However, 17% of participants disagreed with the statement, whilst the rest neither agreed nor disagreed. Due to the lengthy and time-consuming documentation process, several participants suggested that the user should be given a flexibility to capture details, as they desired. In fact, such flexibility has already been incorporated in the proposed system through the tree view of task documents. The user is able to access any task documents in the tree view and record details at any stage of the project. Such an option allows the user to record task details, as desired. Because of the 'modularity' of the task documents, such ad-hoc recording does not affect producing a sensible and meaningful phase-end document. Hence, the length of the documents and the time for producing the documents are not the significant issues to be argued in this approach. Although "Too much detail" is another negative opinion that was raised during the survey, the flexibility to produce the desired document defends such a weakness. Nevertheless, as discussed in Sections 3.3.2.2 and 7.1, testing the methodology in different application domain (i.e. action research) makes provision to establish minimal details of the project. Hence, further research is essential to validate the framework rigorously or to identify the minimal and significant task details of simulation projects.

Difficulty in joining different software with the MRD system or the development of different interfaces for different software is another weakness in the proposed system. Undoubtedly, this issue is the complex matter to be addressed in this approach. However, according Pratt et al., (1994), the use of reusable model elements creates a more convenient environment to develop such interfaces. Nevertheless, such provision should be tested further by developing interfaces for different software. Hence, further research is essential. Although the development of interfaces is a significant issue for existing simulation software, implementation of a standard MRD system inevitably encourages the future software developers to include built-in interfaces for documentation.

The above limitations have been recognized in the study and counteractive actions were taken to avoid or minimize them. However, inability to track changes in project details is the imperative weakness that was not realised before. The proposed approach does not present the details that represent the gradual development of the project. Rather, it displays the ultimate descriptions of the project details. For example, alterations made to the model step-by-step are not displayed in the documentation model. Instead, the documentation model may display one of the instances of the final model that may have been used in experimentation. A remedy to such a weakness is crucial due to iterative nature of model

development process. Hence, further research work is mandatory to record and track the step-by-step development of the project and their details.

7.6 Validity of the Proposed Methodology

In section 7.5, a detailed description of the proposed methodology and their appropriateness, usefulness, advantages, disadvantages, and limitation have been evaluated. The overall results indicate that the proposed methodology has provided a valuable contribution to the documentation in simulation. It has addressed the barriers for not documenting details or varying attempt to documentation. Subsequently, experts are of the opinion that the proposed methodology is a structured and integrated method for the continuous recording of project details in order to present documents with a common layout for different audiences. However, the length, time, and inability to track changes are against the positive views of the methodology. Integration of the documentation process with the model development is the major positive point of this methodology. From their experience in different domains and different software, the respondents are in the favour of implementation of a standard and integrated documentation system. However, software-independent model representation and the development of individual interfaces have been questioned due to the diverse structure of the existing simulation software and different approaches to modelling process. Several participants have shown their natural interest even after the survey was completed. Therefore, it seems that the concept has been conveyed properly and accurately and, hence, the validation process is considered as successful.

7.7 Summary

At the beginning of this chapter discussed the importance of validating of this methodology and presented various strategies for doing so. Then, type, format, boundaries, and limitation of the strategies (questionnaire and case study) were discussed in order to present an overview of the validation process. As the questionnaire was the major strategy in the validation process, the objectives of the questionnaire were presented in line with the sequential development of this methodology. The results of the survey were analysed, followed by detailed discussion on the analysis. In the analysis, the results were evaluated, emphasizing key points in the sequential development of the proposed methodology. Appropriateness, effectiveness, viability, practicability, and weaknesses of the methodology were discussed in great-detail. At appropriate points in the discussion, where weaknesses were raised, remedial actions were suggested. Finally, the concluding remarks were presented in respect of the overall validity of the proposed methodology and the validation process. Conclusion about the proposed methodology is left for discussion in Chapter 08.

8. DISCUSSION, CONTRIBUTION, CONCLUSION AND FURTHER WORK

8.1 Introduction

This chapter first intends to discuss the proposed methodology for Model Representation and Documentation (MRD) in the light of objectives that have been set against the research questions in Section 3.2. During the discussion, it intends to summarize and correlate the research process and the findings, emphasizing the advantages and disadvantages of any approach proposed or technique developed to accomplish the aim of the research. In appropriate points in the discussion, the limitations of the findings are discussed and the avenues for further research are proposed. Having summarized the proposed methodology, the contribution made to the knowledge of simulation is presented, followed by the conclusion of the research. Finally, further research works required for the development of the proposed methodology and for implementation of the methodology are suggested.

8.2 Discussion

8.2.1 Research Question 01 and Model Representation and Documentation

The first research question is ‘What is documentation in simulation and why is it necessary?’ The answer to both part of the question is simply the new notion - ‘Model Representation and Documentation (MRD)’. The following sections briefly outline why a new notion is necessary for documentation in simulation and what the MRD is.

- **MRD to Replace Documentation in Simulation**

Documentation in simulation is interpreted in different contexts and perspectives in the literature. Mostly, documentation refers to model representation. Although the model representation is considered as a vital activity, it is one among many significant components or sets of details of the simulation project. Such varying perspectives in documentation have led to a fresh notion, and therefore the documentation in simulation is named as ‘MRD’ to avoid misconception and to broaden the meaning.

- **Needs of Audiences to find out the Purposes of Documentation**

A part of the first research question aimed to identify the reasons for documentation in simulation. This was accomplished by identifying potential beneficiaries (audiences) from documentation in respect of the simulation project (‘model users’), future use of the model (‘model re-users’), and other interested parties. The analysis of the needs of these individual audiences has then directed the study to identify the purposes of documentation in simulation. This has subsequently led to a new definition for documentation in simulation -MRD- and to answer the first question comprehensively.

- **New Notion for Documentation: MRD**

Having taken the needs of different audiences into account, MRD is defined as recording the process and results of simulation projects for communicating the status of the project, disseminating the accumulated knowledge, motivating the team, ensuring the quality of the project and model, reporting the progress, maintaining the project progress and error, protecting contingency losses and enhancing model reuse. Thereby, MRD provides the answers to both parts of the first research question. The research process performed and any methodology proposed were based on this new notion.

8.2.2 Research Question 02 and Task Documents

The second research question is ‘What are the details that need to be documented in simulation projects?’ The answer to this question is simply ‘task documents’. The following sections briefly outline; what a task document is, how such a conclusion was drawn, and what benefits are offered by the task documents. However, it is not intended to present the findings (i.e. project details) of the research process. Instead, any strategies employed to draw such conclusion are discussed in conjunction with the MRD framework.

- **Task-Orientation for MRD Process**

‘Task-orientation’ (more specifically ‘task documentation’) is the most appropriate approach to fulfil the purposes of MRD. With task-documentation, the process and the results of each task of the simulation project are recorded and stored as the project progresses. The output from task documentation is a group of ‘task details’ and is referred to as ‘task documents’. Task documents are then accumulated and arranged in appropriate forms to fulfil the different purposes of documentation.

- **Task Documentation to Identify Details of Simulation Projects**

The details to be documented primarily depend on the purpose of the document and for whom it was prepared. Due to the diverse range of details demanded for different purposes of different audiences, continuous recording of project details or the task documentation is the most appropriate option available for identifying the details of simulation projects. Consequently, the task documents were considered as the details that are to be recorded in simulation projects.

Task documentation should be based on the simulation project procedure. However, there are inconsistencies in breakdowns and their sub-elements in existing simulation project procedures. Even within a specific procedure, it may be hard to find two similar projects with the same practical approach or the same project with similar approach by different individuals. This situation hampers in establishing the task documentation in a

generic framework. Nevertheless, as the conceptual approach to any simulation project remains analogous, the task documentation should follow a standard project procedure to identify task documents and to establish a generic framework for documentation.

- **Revised Simulation Project Procedure as a Standard Practice**

Task documentation raises a need for investigating a standard project procedure. However, as such investigation is beyond the scope; a revised procedure, based on the existing project procedures, is proposed. The proposed revised procedure shows much in common with the structure presented by Robinson (1994). However, documentation is a parallel process in the revised procedure, rather than a single activity like in typical procedures. With the revised procedure, the simulation process is broken down into 04 major phases and then 13 sub-phases in order to facilitate the task documentation.

- **Task Documents**

Task documents are created by breaking the sub-phases of the simulation project procedure into micro-level of tasks that are referred to as 'primary tasks'. However, the some primary tasks deliver more than one task documents in complying with the modularity of task documents, and the recurrence and exchangeability of task details. As a result, 40 task documents were identified to fulfil the purposes of different audiences. The details of task documents were discussed in Chapters 04 and 05.

- **Task Details**

The contents of task documents (i.e. task details) comprise of both theoretical and practical information of simulation projects and they are more biased to manufacturing applications. Task documents are also included with software-independent details from the beginning of the project, for serving different level of audiences, and for making provisions for model exchange, which will be discussed later in the proposed MRD methodology. In reality, as the foremost focus is on the executable model, the structure and the facilities incorporated within the simulation software always have a certain influence on task details. In the proposed framework, such an influence is kept at a minimal level using 'reusable model elements' and keeping a 'documentation model', which is the outcome from model representation, in focus.

- **Reusable Model Elements to Represent Documentation Model**

The documentation model is primarily constructed with reusable model elements (physical, information, and control). They are the model elements that are separated according to the concept proposed by Bhuskute et al. (1992), Pratt et al. (1994), and Delen et al. (1996). However, the concept is extended further to logical and data

elements for making the documentation process an integral activity. Such an extension facilitates not only to capture the model data and logics, which cannot be ascertained with the use of information and control elements alone, but also to separate the model data from model logics. The separation of model elements provides many benefits not only for explicit representation of the documentation model, but also for model development, translation, alterations, reuse etc, which will be discussed later in the proposed MRD methodology.

- **Reusable Model Elements to Enrich Other Task Details**

Due to the major emphasis on the documentation model and its elements in the proposed methodology, the task details, especially in the tasks before the model representation, are grouped in such a way that they reflect same characteristics of the reusable model elements. The grouping of task details in such a way can make the documentation an efficient activity through functional and hierarchical communication. For example, the details on physical elements gathered separately at the beginning are shared with the model representation task for maintaining the consistency of records.

- **Benefits from Task-Orientation and Task Documents**

A micro-level of abstraction of the simulation project procedure (i.e. primary tasks) and prior declaration of the task steps and the results expected from the task, provide insight into the tasks; and guide the user to perform tasks in the correct order without omitting any of them. Such guidelines inevitably accelerate the model development process and enhance the successfulness of the project. In contrast, recording major and minor details of each task seems to be lengthier and time-consuming than the usual practice. This weakness is addressed in the proposed system (MRD System) by giving flexibility for the user to capture details, as they desired. Even without such flexibility, the benefits from task documentation cannot be undermined in respect of the purposes of documentation. Although, the task documentation is considered as a sequential process, the iterative nature of the simulation project may hamper sequential task documentation. Nevertheless, the modularity of task documents and recurrence of task details provide the solution to such a drawback and produce meaningful documents.

8.2.3 Research Question 03 and Integrated MRD System

The third research question is ‘How the documentation of simulation projects should be accomplished?’ ‘Integrated MRD system’ is the straight answer to this question. The task documentation and the task documents establish the preliminary structure of the documentation process (MRD process) and the documentation system (MRD system). However, according to the objectives of this question, the documentation process should be

an integral activity of simulation projects to make the documentation being popular. An integrated MRD system fulfils such a requisite. Therefore, the following sections briefly outline how the proposed system was established, what facilities it was included, how it improves the present situation in documentation, how it helps the modellers for documentation, what additional benefits it delivers and what level of standard it displays.

- **How should the Documentation of Simulation Project be accomplished?**

Non-availability of structured documentation methodology with guidelines is the foremost barrier for poor practice in documentation in simulation. Although retrospective documentation is a less significant barrier, it hinders the documentation due to the tendency to forget details. These problems are addressed by structured task-oriented approach that makes provision not only for structuring the task documents, but also for recording the project details progressively with the model development process. However, task documentation and model development are still isolated processes and run as two distinct processes. Such an isolated documentation process is another barrier in the current practice and makes the documentation unpopular. Therefore, they are integrated to execute them as a single process, benefiting for both processes directly and mutually. Such an integrated MRD system inevitably improves the documentation. Modellers, then consider documentation as an integral activity in simulation projects. In the proposed system, such an environment is accomplished by:

- maintaining a uniform and common structure for documents in any project,
- sharing data among task documents for an efficient documentation system,
- exchanging model between two processes for an un-isolated MRD system, and,
- interfacing with the user for a comprehensible documentation system.

The above listing shows neither the features incorporated nor the advantages in the proposed methodology. Therefore, they are briefly discussed below.

8.2.3.1 Uniform Structure

Uniformity of the documentation structure is maintained by introducing pre-structured forms for task documents, documentation templates for producing documents with a common layout, data tables to present the details that need sharing, and a reusable model elements library to represent the model independently from the simulation software.

- **Pre-structured Forms for Uniform Task Recording**

Pre-structured forms, designed for almost all task documents, maintain the clarity, consistency, and uniformity of details for any simulation project. Such forms serve as a checklist for the project team to be aware of the tasks that need to be performed and the outcomes from each task. Furthermore, they enable even the non-expert or the supporting staff to contribute towards the documentation.

- **Documentation Templates as a Common Layout**

Documentation templates are embedded with pre-structured forms of task documents. The user could use the templates to record the details of task documents and produce the phase-end documents of different projects with a common layout. Due to the popularity, available interfaces and applications in other related fields, MS Word is considered as the most appropriate software to create documentation templates. Hence, the user can use a familiar environment to record task details and produce documents for different needs. However, templates need interaction with other software such as Visio and Access to create and retrieve task details that are in different formats.

- **Elements Library for Documentation Model**

Reusable model elements jointly describe the interaction of system objects, represent the logics of the executable model, and capture model data. These elements are composed into a documentation model element library – stencil – in generic form, and are embedded with user properties. Most of the properties are collected at the time of using the elements as user entries. Some properties are the identities that have already been defined. Some are placed interacting with other properties in the same element or other elements. The properties are primarily used to present the text description of elements for the purpose of documentation. Thereby, this approach maintains the uniformity and consistency of the model description throughout any project. The same properties are employed for translating the documentation model into executable model through a common database, which will be discussed later. MS Visio software is appropriate to create the stencil, as it makes provision to capture attributes of individual elements and to link shapes/diagrams with external databases (e.g. common database).

- **Data Tables for Presenting and Sharing Details**

Undoubtedly, MS Word is passable to record the model data. However, for an integrated documentation environment, where the documentation model and model data are translated into the executable model, model data is recorded as MS Access tables with pre-defined fields and formatted details. The use of data tables in Access makes a provision to interface them with Visio and to create a common database.

8.2.3.2 Sharing Data

The MRD system is primarily associated with different subsystems. Subsystems are created with the use of templates in different software environment. The user is able to utilize the templates and work with new files to capture details of new projects. In recording the details, the user may need to repeat or share the same data in different task documents in compliance with the modularity of task documentation. Data is shared through functional or

hierarchical communication between different subsystems or different levels of the same subsystem. Such communication is accommodated through the user interfaces, developed with VB or VBA as described in Chapter 06. For example, the details of physical elements, recorded previously, are used to construct the documentation model. Similarly, the elements in the documentation model are utilized to prepare elements list for recording the verification details. Therefore, data sharing makes the MRD system more efficient.

8.2.3.3 Exchanging the Model

Although the documentation is successful with the proposed uniform structure and data sharing, it is still an isolated process, unless it is integrated with the model development process. Such integration is achieved by translating the documentation model (DM) into an executable model, and updating the DM to capture alterations that were made to the executable model. However, model exchange needs creation of an appropriate environment. Such an environment is established through neutral representation of the DM, construction of the DM in comparable with the executable model and development of a common database. Therefore, the following sections discuss how such environment is established.

- **Documentation Model as a Neutral Representation**

If the elements used for the construction of the DM is software-independent and the method adopted to describe the model is a neutral technique, then the DM is considered as a generic representation of the model, which makes provision to translate the DM into a software-dependent executable model. The DM is constructed with the use of neutral elements, which are based on the extended separation concept. The flowchart technique is adopted to construct the DM, as it is neutral and offers many more benefits compared with other model representation techniques in respect of documentation. Nevertheless, the use of neutral model elements and a neutral technique with its stipulated symbols may reflect a conflict in application. However, the ability to customize the flowchart symbols to fit into any need avoids such a conflict. Therefore, amalgamation of flowchart symbols and its features with neutral model elements makes a provision to customize the documentation model elements. Such an amalgamation reflects the development of VIM systems and GU interfaces, where model elements are simplified to represent different system features.

- **Documentation Model in Comparable with Executable Model**

The construction of DM follows the same approach that is generally used for the development of the executable model, in which the complexities are added gradually to ensure the development of the right model. DM elements offer a similar environment to construct the DM and abstract information in line with the modeller's natural thinking.

- **Documentation Model Database as a Common Database**

Even with neutral model elements and technique, the DM cannot be translated without capturing the details of individual elements and making a database of the DM as medium of model transfer. The user properties embedded in model elements and the data produced interacting with other elements are used to create the DM database. DM elements are reusable and perform one-to-one function of the model, and their properties are software-independent. Therefore, the DM database, which has a standard internal structure to store those model details, is considered as a common database. Such a database makes a provision for model translation with potential software interfaces. The feasibility of the common database was studied with Arena for process-related models. It proved that a model could be translated with the same data utilized for the purpose of documentation. Hence, the documentation tends to an un-isolated process. However, with the present development, a certain modification such as adding connections, capturing built-in variables, etc. are necessary to run the translated model.

8.2.3.4 Interfacing with the User

A structured, progressive, software-independent, and integrated documentation system is the key to overcoming the present poor attempt for documentation in simulation. For such a system, the user needs to be guided through the documentation process, interacting with different software to create, store, retrieve, share, and exchange the task details. With those capabilities, the documentation process is considered as fully or partially automated with the model development. Such automation is attained through VB and VBA interfaces that manage and control the documentation environment effectively. In the proposed framework, such an automated system is referred to as MRD Tool. As described in Chapter 06, the following features are embedded in the MRD tool to provide appropriate assistance.

- Menu structures to accommodate a new project and to alter an existing document.
- Expandable tree view with check boxes to display task documents and differentiate between the task documents that have already been recorded and not been recorded.
- Access task documents at any stage to store and retrieve details in different software.
- Immediate display screen to display the contents of individual task documents.
- Bold or highlight of task documents to give attentions that require alterations.
- Options for the user in functional and hierarchical communication.
- Guidelines (messages) to perform the appropriate task at right time and right direction.
- Export (pre-documentation) and update (post-documentation) facilities for translating the DM and capturing alterations made to the executable model.
- Facilities to prepare documents at any stage for various needs, with users' choice.

8.2.3.5 Limitations and Weaknesses in MRD System

Within the present development, the MRD system (or MRD Tool) performs only a very few functional and hierarchical communications among the subsystems as well as in different levels of the same subsystem. Also, the documentation model is limited to process-related models and the interface for model exchange is confined to Arena software only. Post-documentation is limited to updating shapes and their identities only, as it was assumed that a detailed update may not be necessary due to the proposed extensive modelling approach.

The above restrictions are mainly due to the limited development of the MRD tool. However, there are weaknesses in the conceptual approach, as discussed in Section 7.5.6. The length of the document, time to complete the document and excessive details are considered as general weaknesses in this approach. Nevertheless, as discussed in the validation process, the flexibility incorporated in the MRD system to capture details as the user desired and to produce the desired document, defend such weaknesses. The foremost weakness is the inability to record and track the step-by-step development of the project and their details. This was not realized during the study and therefore no counteractive actions were taken. Hence, further research work is mandatory to capture such details. Although the development of interfaces for different software is considered as a weakness, no alternative is available to avoid such development due to varying structure of software.

8.2.3.6 MRD System as a Stand-Alone or Remote System

Although the templates or subsystems are interfaced in order to create an integrated documentation environment, the templates alone can serve as a remote or a stand-alone documentation system. Task details can be recorded directly with the use of document templates and the DM can be constructed with the use of the DM stencil, but without data sharing and model exchange. However, a detailed construction of DM and an extensive approach for recording task details, as proposed in this methodology, may not be worthwhile, though such a remote documentation serves the purpose of documentation.

8.2.4 Additional Benefits from MRD System

In addition to the benefits discussed previously in respect of documentation, the MRD system also offers other benefits to simulation community in respect of the following.

- **MRD System as a Project Management and Control System**

The MRD system guides the project team to record task details and provides the necessary information for their day-to-day management and operational tasks, like conveying tasks to be performed, individual responsibilities and time scales, reporting progress, etc. to ensure the success of the project. As a result, it acts as a management and control system for simulation projects.

- **MRD System as a Training System**

The MRD system helps the user, not only to record project details, but also to perform simulation projects step-by-step with necessary advice and guidelines. Therefore, it assists the novice practitioner as to what and how tasks are performed, to ensure the success of simulation projects. As a result, it acts as a training tool to a novice user.

- **MRD System to Enhance the Model Reusability**

It is a fact that any form of documentation increases the model reusability as it conveys the details of what went into constructing the model. The DM and common database in the proposed MRD system further enhance the model reusability as they act as the media for exchanging the model among the different software systems.

8.2.5 MRD System or Standard MRD System

The successful development and implementation of a documentation system is contingent, to a great extent, on the establishment of sound documentation standards that specify a common layout and systematically linked and interfaced details, which result from each task of the simulation project. Such attributes are addressed in the MRD system. However, the following issues that were raised during the study should further be explored for a standard MRD system.

- Need for a standard project procedure and terms with their definitions for task documentation are the issues to be investigated further, though they were addressed with the revised project procedure, largely dealing with the Robinson (1994) approach.
- Needs for defining standard and formalized model elements to create a generic DM and a common database, is still a remaining issue, as the proposed elements may have been influenced by the author's experience with Arena software.
- Notations such as standard rules (e.g. FIFO) and format [e.g. TRIA (Min,Mode,Max)] used in the MRD tool are based on Arena software. Hence, establishing standard notations are still a matter to be researched.

The above issues were not investigated comprehensively in the proposed system. Therefore, they should be explored further to improve the proposed methodology and to establish documentation standards.

- **Documentation Standards**

The answers to the above issues will improve documentation standards further to address the model exchangeability and reusability. With such complete documentation standards, the MRD system will be considered as a standard MRD system. It will offer more benefits (besides what was described before) to the simulation community as it

attempts to establish a generic framework that enables the modeller to employ it as a management and control system for simulation projects, while the documentation is being focused. This would inevitably allow access for non-experts to the simulation community. Fig. 7.1 illustrates the benefits and the integration of the MRD process and the simulation project through such documentation standards.

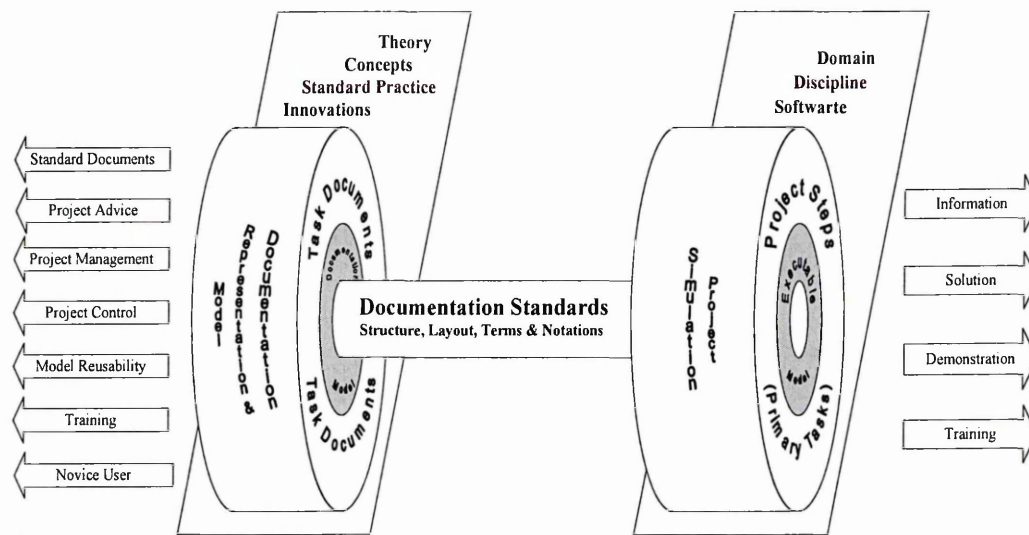


Fig. 7.1 : Standard MRD Process and Simulation Project

As can be seen in the figure (7.1), when the simulation project begins to roll over the different domain and different simulation software, the MRD begins to turn over the simulation fundamentals for the purposes of documentation. If the two processes are coupled or integrated through the documentation standards, then the two processes could be considered as a single process, which will offer more benefits to the simulation community.

- **Do the Documentation Standards Hamper the Creativity?**

There is an argument that standardization may hamper the creativity and slowdown the introduction of innovations. In simulation applications, such argument may be invalid, as the standard MRD system presents only the framework, leaving the contents of the application unchanged. Moreover, the use of simulation software itself is bound into its own framework rather than a generic framework as proposed with the MRD system. In this respect, adhering to a generic framework will be an added advantage together with other benefits as discussed before. In respect of new concepts, such standardization may have a negative influence on creativity and innovations. However, if the MRD framework is flexible to accommodate innovations, then such standardization is not an issue to be resisted.

8.3 Contribution

In terms of contribution, the following major contributions were made by this research to the knowledge in the field of simulation.

- **Model Representation and Documentation (MRD) Process**

Model Representation and Documentation (MRD) is a new concept for documenting simulation projects. It fulfils the needs of different audiences in respect of the simulation project, potential future use of the model and other interested parties. The MRD process is based on task-orientation or, more specifically, on task documentation. Task documentation offers a sound environment to capture project details progressively with the model development process and to structure the documents and the documentation process for a successful implementation of a documentation system.

- **Integrated MRD System**

The MRD process with structured task documents that are enriched by reusable model elements, inherit to the development of MRD system. The MRD system, which specifies a common layout and systematically linked details that result from each task of the simulation project, enables the user to record project details efficiently through functional and hierarchical integration. The integrated MRD system offers an environment for the user to perform both documentation and model development concurrently while both are benefited directly and mutually through model exchange. This, inevitably, not only improves documentation of simulation, but also manages, controls and monitors the progress of simulation projects.

- **Documentation Model (DM) and Common Database**

The Documentation Model (DM) is a generic representation of the executable model. The use of amalgamated reusable model elements and a neutral flowcharts technique to construct the DM makes a provision to translate the DM into a software-dependent executable model. The common database that has a standard internal structure to store the details of the DM acts as a media for model exchangeability and reusability.

- **Bridging Computer and Practical Simulation Project Life Cycles**

The use of reusable model elements is a computer science aspect related to the model development. However, such a use for the purpose of documentation is a practical aspect of the simulation project. The documentation model and common database that comprise of documentation standards link those two aspects. Therefore, development of the integrated MRD system can be considered as an attempt to bridge the computer simulation project life cycle and the practical simulation project life cycle.

8.4 Conclusion

The research process that has led to the development of a methodology to document simulation projects in manufacturing applications can be concluded as follows:

- Documentation in simulation involves recording the project details of what went into constructing the model, incorporating the modellers' knowledge and experience; and communicating and making them accessible to others who may be interested.
- Without of a structured methodology or guidelines for producing simulation project documents and the separate documentation process results in the documentation of simulation projects being unpopular, ignored or given little prominence.
- Poor documentation prevents others from understanding of simulation models, the broadening of model reusability, and spreading of simulation concepts.
- Progressive documentation that is based on task-orientation and incorporates the simulation basic concepts, offers a strategy, not only to fulfil the needs of audiences in the project and future usage of the model, but also to establish a framework for documentation in simulation.
- Amalgamated reusable model elements with neutral flowcharts technique, not only describes the model explicitly and independently from simulation software, but also make provision for model exchange with the simulation software.
- The integration of the documentation process and the model development process, not only improves documentation of simulation projects, but also manages, controls and monitors the progress of simulation projects.
- The development of a common database to store the model descriptions, based on documentation standards offers an avenue for enhancing the model reusability.
- A word of caution: Necessity of documentation standards to create standard documentation system is compulsory, as the benefits to be accrued from standardization are considered high in respect of both simulation applications and innovations.

8.5 Further Work

The proposed methodology to document simulation projects is still at the development stage. It needs further work to put it into standard practice. This further work could focus on three perspectives: validation and evaluation, standardization and implementation.

- Although the methodology was validated through a questionnaire survey, action research is considered as the most appropriate option for an elaborate empirical evaluation in a real-world environment that would enable a statistically valid conclusion to be drawn. Such a rigorous validation is yet to be done. Therefore, the proposed methodology needs applying for real word problems that represent different scenarios in diverse disciplines, for making the appropriate alterations to the framework and subsequently to present a more realistic solution to documentation in simulation.
- Within the present development, model exchange is permitted to Arena (Version 5) software only. Even with Arena, it demonstrates the feasibility limited to process-related systems. To evaluate the validity of the methodology, model exchange should be extended to other Arena modules as well as other simulation software, representing both procedural and object-oriented paradigms, with potential development of appropriate interfaces.
- In extending the model exchange to other software, it inevitably elevates the issue of standardization in respect of creating and simplifying generic model elements and establishing common terms and notations. Such issues should be addressed broadly and universally to establish common documentation standards.
- With the development of sound documentation standards, a documentation system could be implemented as stand-alone commercial software or a separate system module attached to individual simulation software. Although a necessity of built-in individual interfaces for different software would arise in the case of stand-alone system, such interfaces inevitably enhance the model-reuse. Further, web-based documentation may be a better prospect for stand-alone system due to convenient access and exchange of project details, if confidentiality of information and data are sustained and can be guaranteed.

REFERENCES

- Abrams, M.D. and M. V. Zelkowitz. 1995. Striving for correctness. *Computers & Security*, vol. 14, 719-738.
- Al-Ahamari, A.M.A. and K. Ridgway. 1999. An integrated modelling method to support manufacturing system analysis and design. *Computers in Industry*, vol. 38, no. 3, 225-238.
- Anderson, M. and G. Olsson. 1998. 1998. A simulation based decision support approach for operational capacity planning in a customer order driven assembly line. In *Proceedings of the 1998 Winter Simulation Conference*, ed. J. S. Carson, M. S. Manivannan, D. J. Medeiros, and E. F. Watson. 935-941. Washington DC, USA.
- Anglani, A., A. Grieco, M. Pacella, and T. Tolio. 2002. Object-oriented modelling and simulation of flexible manufacturing systems: a rule-based procedure. *Simulation Practice and Theory*, vol. 10, 209-234.
- Arons, H.de.S. and E. Van Asperen. 2000. Computer assistance for model definition. In *Proceedings of the 2000 Winter Simulation Conference*, ed. P. A. Fishwick, K. Kang, J. A. Joines, and R. H. Barton. 399-408. Orlando, FL, USA.
- Ayer, S. and F. Patrinostro. 1992. *Documenting the Software Development Process*, McGraw-Hill, Inc., New York.
- Balci, O. 1997. Verification, validation, and Accreditation of Simulation Models. In *Proceeding of the 1997 Winter Simulation Conference*, ed. D. H. Withers, B. L. Nelson, S. Andradottir, and K. J. Healy. 135-141. Atlanta, GA, USA.
- Balci, O. 1998. Verification, validation, and Accreditation. In *Proceeding of the 1998 Winter Simulation Conference*, ed. J. S. Carson, M. S. Manivannan, D. J. Medeiros, and E. F. Watson. 41-48. Washington DC, USA.
- Banks, J. 1992. Simulation languages and simulators. In *Proceedings of the 1992 Winter Simulation Conference*, ed. R. C. Crain, J. R. Wilson, J. J. Swain, and D. Glodsman. 88-96. Arlington, VA, USA.
- Banks, J. 1996. Software for simulation. In *Proceedings of the 1996 Winter Simulation Conference*, ed. J. M. Charnes, D. J. Morrice, D. T. Brunner, and J. J. Swain. 31-38. Coronado, CA, USA.
- Banks, J. 2000. Introduction to simulation. In *Proceedings of the 2000 Winter Simulation Conference*, ed. P. A. Fishwick, K. Kang, J. A. Joines, and R. R. Barton. 9-16. Orlando, FL, USA.
- Barker, T.T. 1998. *Writing Software Documentation*, Allyn & Bacon.
- Basili, V.R. and S. K. Abd-Ei-Hafiz. 1996. A method of documenting code components. *Journal of System Software* vol. 34, 89-104.
- Benjamin, P., D. Delen, R. Mayer, and T.O'Brien. 2000. A model based approach for component simulation development. In *Proceedings of the 2000 Winter Simulation Conference*, ed. P. A. Fishwick, K. Kang, J. A. Joines, and R. R. Barton. 1831-1839. Orlando, FL, USA.
- Benz, J, Hoch, R. and T. Gabele. 1997. Documentation of mathematical models – Ecology – an unpopular task?. *Ecological Modelling*, vol. 1, no. 8, 1-8.
- Benz, J. and M. Knorrenschild. 1997. Call for a model documentation etiquette, *Ecological Modelling*, vol. 97, 141-143.

- Benz, J., Hoch, R. and T. Legovic. 2001. ECOBAS: modelling and documentation, *Ecological Modelling*, vol. 138, 3-15.
- Bhuskute, H.C., M.N. Duse, J. T. Gharpura, D. B. Pratt, M. Kamath, and J. H. Mize. 1992. Design and implementation of highly reusable modelling and simulation framework for discrete part manufacturing systems. In *Proceedings of the 1992 Winter Simulation Conference*, ed. R. C. Crain, J. R. Wilson, J. J. Swain, and D. Glodsman. 680-688. Arlington, VA, USA.
- Bortscheller, B. J. and E. T. Saulnier. 1992. Model reusability in graphical simulation package. In *Proceedings of the 1992 Winter Simulation Conference*, ed. R. C. Crain, J. R. Wilson, J. J. Swain, and D. Glodsman. 764-772. Arlington, VA, USA.
- Brooks, J.B. and S. Robinson. 2001. *Simulation and inventory control*, Hampshire: Palgrave.
- Buss, A.H. 2000. Component-based simulation modelling. In *Proceedings of the 2000 Winter Simulation Conference*, ed. P. A. Fishwick, K. Kang, J. A. Joines, and R. H. Barton. 964-971. Orlando, FL, USA.
- Carson, J.S. 1992. Modelling. In *Proceedings of the 1992 Winter Simulation Conference*, ed. R. C. Crain, J. R. Wilson, J. J. Swain, and D. Glodsman. 82-87. Arlington, VA, USA.
- Castano, S. and V. De Antonellis. 1997. Engineering a library of reusable conceptual components. *Information and Software Technology*, vol. 39, 65-79.
- Centeno, M. A. 1996. An introduction to simulation modelling. In *Proceedings of the 1996 Winter Simulation Conference*, ed. J. M. Charnes, D. J. Morrice, D. T. Brunner, and J. J. Swain. 15-22. Coronado, CA, USA.
- Ceric, V. and R. Paul. 1994. Diagrammatic representation of the conceptual model. *Mathematics and Computers in Simulation*, vol. 34, 317-324.
- Chen, G. and B. K. Szymanski. 2001. Component-oriented simulation architecture: towards interoperability and interchangeability. In *Proceedings of the 2001 Winter Simulation Conference*, ed. M. Rohrer, D. Medeiros, B. A. Peters, and D.J. Smith. 495-501. Arlington, VA, USA.
- Chen, G. and B. K. Szymanski. 2002. COST: A component-oriented discrete event simulation. In *Proceedings of the 2002 Winter Simulation Conference*, ed. J. L. Snowdon, J. M. Charnes, E. Yucesan, and C. Chen. 776-782. San Diego, CA , USA.
- Chu, P. C. 1997. Actors and scripts: object-oriented simulation of enterprise system. *System Software*, vol. 37, 187-199.
- Clark, G. M. 1996. An introduction to manufacturing applications. In *Proceedings of the 1996 Winter Simulation Conference*, ed. J. M. Charnes, D. J. Morrice, D. T. Brunner, and J. J. Swain. 85-92. Coronado, CA, USA.
- Daum, T. and R. G. Sargent. 1999. Scaling, hierarchical modelling and reuse in an object-oriented modelling and simulation systems. In *Proceedings of the 1999 Winter Simulation Conference*, ed. D. T. Sturrock, G. W. Evans, P. A. Farrington, and H. B. Nemhard. 1470-1477. Phoenix, AZ, USA.
- Delen, D., D. B. Pratt, and M. Kamath, M. 1996. A new paradigm for manufacturing enterprise modelling: reusable, multi-tool modelling. In *Proceedings of the 1996 Winter Simulation Conference*, ed. J. M. Charnes, D. J. Morrice, D. T. Brunner, and J. J. Swain. 985-992. Coronado, CA, USA.
- Delen, D., P.C. Benjamin, and M. Erraguntla. 1998. Integrated modelling and analysis generator environment (IMAGE): A decision support tool. In *Proceedings of the 1998 Winter*

Simulation Conference, ed. J. S. Carson, M. S. Manivannan, D. J. Medeiros, and E. F. Watson. 1401-1408. Washington DC, USA.

Duse, N. M., J. T. Gharpura, H. Bhuskute, M. Kamath, D. B. Pratt, and J. H. Mize. 1993. Tool-independent model representation. *Proceedings of 2nd Industrial Engineering Research Conference*, Norcross, GA, 700-704, IIE.

Eldabi, T. and R. J. Paul. 1997. Flexible modelling of manufacturing systems with variable levels of details. In *Proceeding of the 1997 Winter Simulation Conference*, ed. D. H. Withers, B. L. Nelson, S. Andradottir, and K. J. Healy. 801-808. Atlanta, GA, USA.

Forward, A. and T. Lethbridge. 2002. The relevance of software documentation, tools and technologies: a survey. *ACM Symposium on Document Engineering 2002*, 26-33. McLean, Virginia, USA. Available from: www.site.uottawa.ca/~aforward/docsurvey/, [26th July 2003].

French, J. C., J. C. Knight, and A. L. Powell. 1997. Applying hypertext structures to software documentation. *Information Processing and Management*, vol. 33, no. 2, 219-231.

Gilbert, N. 2001. *Researching Social Life*. Sage Publications Inc., London.

Gill, J. and P. Johnson. 1991. *Research Methods for managers*. Paul Chapman Publishing.

Gray, M. and K. London. 1969. *Documentation Standards*, 1st edn., Random Systems Press, Inc.

Gummesson, E. 1991. *Qualitative Methods in Management Research*, Sage Publications, Inc., London

Hoch, R., T. Gabele, and J. Benz. 1998. Towards a standard for documentation of mathematical models. *Ecological Modelling*, vol. 113, no. 1-3, 3-12.

Huda, A. M. and C. A. Chung. 2002. Simulation modelling and analysis issues for high speed combined continuous and discrete food industry manufacturing processes. *Computers & Industrial Engineering*, vol. 43, 473-483.

Hulpic, V. and R. J. Paul. 1996. Methodological approach to manufacturing simulation software selection. *Computer Integrated Manufacturing Systems*, vol. 9, no. 1, 49-55.

Hung, Y. and R. K. Iyer. 1998. An object-oriented environment for past simulation using compiler techniques. In *Proceedings of the 1998 Winter Simulation Conference*, ed. J. S. Carson, M. S. Manivannan, D. J. Medeiros, and E. F. Watson. 531-538. Washington DC, USA.

Ingalls, R. G. 2002. Introduction to simulation. In *Proceedings of the 2002 Winter Simulation Conference*, ed. J. L. Snowdon, J. M. Charnes, E. Yucesan, and C. Chen. 7-16. San Diego, CA, USA.

Joines, A. J. and S. D. Roberts. 1996. Design of object-oriented simulation in C++. In *Proceedings of the 1996 Winter Simulation Conference*, ed. J. M. Charnes, D. J. Morrice, D. T. Brunner, and J. J. Swain. 65-72. Coronado, CA, USA.

Joines, A. J. and S. D. Roberts. 1998. Fundamentals of object-oriented simulation. In *Proceedings of the 1998 Winter Simulation Conference*, ed. J. S. Carson, M. S. Manivannan, D. J. Medeiros, and E. F. Watson. 141-149. Washington DC, USA.

Joines, A. J. and S. D. Roberts. 1999. Simulation on object-oriented world. In *Proceedings of the Winter Simulation Conference*, ed. D. T. Sturrock, G. W. Evans, P. A. Farrington, and H. B. Nemhard. 132-140. Phoenix, AZ, USA.

Kateel, G., M. Kamath, and D. Pratt. 1996. An overview of CIM enterprise modelling methodologies. In *Proceedings of the 1996 Winter Simulation Conference*, ed. J. M. Charnes, D. J. Morrice, D. T. Brunner, and J. J. Swain. 1000-1007. Coronado, CA, USA.

Keller, R. M. and J. L. Dungan. 1999. Meta modelling: a knowledge-based approach to facilitating process model construction and reuse. *Ecological Modelling*, vol. 119, no. 2-3, 89-116.

Kelton, W. D. 2000. Experimental Design for Simulation. In *Proceedings of the 2000 Winter Simulation Conference*, ed. P. A. Fishwick, K. Kang, J. A. Joines, and R. H. Barton. 32-38. Orlando, FL, USA.

Kelton, W. D., R. P. Sadowski, R.P. and D. A. Sadowski. 2002. *Simulation with Arena*, McGraw-Hill Co., Inc., New York.

Kienbaum, G. and R. J. Paul. 1994. H-ACD: hierarchical activity cycle diagrams for object-oriented simulation modelling. In *Proceedings of the 1994 Winter Simulation Conference*, ed. D. A. Sadowski, A. F. Seila, J. D. Tew, and S. Manivannan. 600-610. Orlando, FL, USA.

Kim C. H., R. H. Weston, A. Hodgson, and K. H. Lee. 2003. The complementary use of IDEF and UML modelling approaches. *Computers in Industry*, vol. 50, 35-56.

Kiran, A.S., T. Centinkaya, and J. Cabrera. 2001. Hierarchical modelling of a shipyard integrated with an external scheduling application. In *Proceedings of the 2001 Winter Simulation Conference*, ed. M. Rohrer, D. Medeiros, B. A. Peters, and D.J. Smith. 877-881. Arlington, VA, USA.

Kleijnen, J. P. C., A. V. Noordedegraff, and M.Niele. 2001. Sensitivity Analysis of Censored Output through Polynomial, Logistic, and Regression Meta-Models: Theory and Case Study. In *Proceedings of the 2001 Winter Simulation Conference*, ed. M. Rohrer, D. Medeiros, B. A. Peters, and J. Smith. 486-491. Arlington, VA, USA.

Klingstam, P. and P. Gullandder. 1999. Overview of simulation tools for computer aided production engineering. *Computers in Industry*, vol. 38, 173-186.

Knowledge Based Systems, Inc. (KBSI). 1995. IDEF- Integrated computer aided manufacturing DEFinition. Available from: <http://www.idef.com>. [29th July 2003].

Knowledge Based Systems, Inc. (KBSI). 1999. IDEF0- Functional modelling method. Available from: <http://www.idef.com/idef0.html> [26th July 2002].

Knowledge Based Systems, Inc. (KBSI). 2000. IDEF3- Process and object state description capture method overview. Available from: <http://www.idef.com/idef3.html> [29th July 2003].

Kolb, D. A., I. M. Rubin and J.M. McIntyre. 1984. *Organizational psychology: an experimental approach to organizational behaviour*, Prentice Hall, London.

Koskossidis, D. A. and C. J. Brennan. 1994. A review of simulation techniques and modelling. In *Proceedings of the 1994 Winter Simulation Conference*, ed. D. A. Sadowski, A. F. Seila, J. D. Tew, and S. Manivannan. 55-58. Orlando, FL, USA.

Kovacs, G. L., S. Kapacsi, J. Nacs, G. Haidegger, and P. Groumpos. 1999. Application of software reuse and object-oriented methodologies for the modelling and control of manufacturing systems, *Computers in Industry*, vol. 39, no. 3, 177-189.

Kwon, O. B. and S. J. Park. 1996. RMT: A modelling support system for model reuse. *Decision Support Systems*, vol. 16, 131-153.

Lanner, 2002. Witness simulation software. <http://www.lanner.com>. [10th June 2003]

Laudon, K. C. and J. P. Laudon. 1978. *Management information Systems*, Prentice Hall, Inc, New Jersey.

- Law, A. M. and M. G. McComas. 1998. Simulations of manufacturing systems. In *Proceedings of the 1998 Winter Simulation Conference*, ed. J. S. Carson, M. S. Manivannan, D. J. Medeiros, and E. F. Watson. 49-52. Washington DC, USA.
- Law, A. M. and W. D. Kelton. 1991. *Simulation modelling and analysis*, 2nd edn., McGraw-Hill Inc., USA.
- Liyanage, K. 1999. *Methodology for input data modelling in the simulation of manufacturing systems*, An unpublished PhD thesis, Sheffield Hallam University, Sheffield.
- Locke, L. F., S. J. Silverman and W. W. Spirdoso. 1998. *Reading and Understanding Research*, Sage Publications Inc.
- Manansang, H. and J. A. Heim. 1996. An on-line simulation-based patient scheduling systems. In *Proceedings of the 1996 Winter Simulation Conference*, ed. J. M. Charnes, D. J. Morrice, D. T. Brunner, and J. J. Swain. 1170-1175. Coronado, CA, USA.
- Maria, A. 1997. Introduction to modelling and simulation. In *Proceeding of the 1997 Winter Simulation Conference*, ed. D. H. Withers, B. L. Nelson, S. Andradottir, and K. J. Healy. 7-13. Atlanta, GA, USA.
- Matko, D., B. Zupancic, and R. Karba. 1992. *Simulation and modelling continuous systems*, Prentice Hall International (UK) Ltd.
- McLean, C., A. Jones, T. Lee, and F. Riddick. 2002. An architecture for a generic data driven machine shop simulator. In *Proceedings of the 2002 Winter Simulation Conference*, ed. J. L. Snowdon, J. M. Charnes, E. Yucesan, and C. Chen. 1108-1116. San Diego, CA , USA.
- Mertins, K., M. Rabe, and F. W. Jaekel. 2000. Neutral template libraries for efficient distributed simulation within a manufacturing system engineering platform. In *Proceedings of the 2000 Winter Simulation Conference*, ed. P. A. Fishwick, K. Kang, J. A. Joines, and R. H. Barton. 1549-1557. Orlando, FL, USA.
- MS Visio. 2002. Overview of Visio product: <http://www.microsoft.com/office/preview/visio/overview.asp> [26th August 2003]
- Muller, D. J. 1996. Simulation: what to do with the model afterward. In *Proceedings of the 1996 Winter Simulation Conference*, ed. J. M. Charnes, D. J. Morrice, D. T. Brunner, and J. J. Swain. 729-733. Coronado, CA, USA.
- Musselman, K. J. 1994. Guidelines for simulation project success. In *Proceedings of the 1994 Winter Simulation Conference*, ed. D. A. Sadowski, A. F. Seila, J. D. Tew, and S. Manivannan. 88-95. Orlando, FL, USA.
- Nance, R. E. 1979. Model representation in discrete event simulation: Prospect for Developing Documentation Standards. Current issues in simulation in (chapter 6). Academic Press, Inc, 83 -97.
- Neelamkavil, F. 1988. *Computer simulation and modelling*. John Wiley & Sons Ltd., GB.
- Oakshott L. 1997. *Business modelling and simulation*, Pitman Publishing, London.
- Odhabi, H. I., R. J. Paul, and R. D. Macredie. 1998. Developing a graphical user interface for discrete event simulation. In *Proceedings of the 1998 Winter Simulation Conference*, ed. J. S. Carson, M. S. Manivannan, D. J. Medeiros, and E. F. Watson. 429-436. Washington DC, USA.
- Oscarsson, J. and M. U. Moris. 2002. Documentation of discrete even simulation models for manufacturing system life cycle simulation. In *Proceedings of the 2002 Winter Simulation Conference*, ed. J. L. Snowdon, J. M. Charnes, E. Yucesan, and C. Chen. 1073-1078. San Diego, CA , USA.

OSU-CIM. 1997. Advanced modelling project. <http://www.okstate.edu/cocim/advmod>. [10th August 2003]

Pandya, K. V., A. Karlsson, S. Segal, and A. Carrie. 1997. Towards the manufacturing enterprise of the future, *International Journal of Operations Management*, vol. 17, no. 5, 502-521.

Pandya, K.V. 1995. Review of modelling techniques and tools for decision making in manufacturing management, *IEEE Proceedings*, vol. 142, no. 5, 371-377.

PathMaker. 2002. Flowcharting. <http://www.skymark.com/resources/tools/flowchart.asp>. [26th August 2003]

Paul, R. J. 1993. Activity cycle diagrams and the three-phase method. In *Proceedings of the 1993 Winter Simulation Conference*, ed. E. C. Russell, W. E. Biles, G. W. Evans, and M. Mollaghasemi. 123-131. Los Angeles, CA, USA.

Paul, R. J. and S. J. E. Taylor. 2002. What use is model reuse: Is there a cook at the end of the Rainbow?. In *Proceedings of the 2002 Winter Simulation Conference*, ed. J. L. Snowdon, J. M. Charnes, E. Yucesan, and C. Chen. 648-652. San Diego, CA, USA.

Pegden, C. D., R. E. Shannon, and R. P. Sadowski. 1990. *Introduction to simulation using SIMAN*. System Modelling Corporation, USA.

Perera, T. and K. Liyanage. 2000. Methodology for rapid identification and collection of input data in the simulation of manufacturing systems, *Simulation Practice & Theory*, vol. 7, 645-656.

Pfleeger, S. L. 2001. *Software engineering theory and practice*, 2nd Edn., Prentice Hall, Inc.

Pidd, M. 1994. In introduction to computer simulation. In *Proceedings of the 1994 Winter Simulation Conference*, ed. J. L. Snowdon, J. M. Charnes, E. Yucesan, and C. Chen. 772-775. San Diego, CA, USA.

Pidd, M. 2002. Simulation software and model reuse: A polemic. In *Proceedings of the 2002 Winter Simulation Conference*, ed. D. A. Sadowski, A. F. Seila, J. D. Tew, and S. Manivannan. 55-58. Orlando, FL, USA

Pidd, M., N. Oses, and R. J. Brooks. 1999. Component based simulation on the Web?. In *Proceedings of the Winter Simulation Conference*, ed. D. T. Sturrock, G. W. Evans, P. A. Farrington, and H. B. Nemhard. 1438-1444. Phoenix, AZ, USA.

Pidd, M. 1992. *Computer simulation in management science*, 3rd edn., John Wiley and Sons, Chichester, UK.

Pidd, M. 1996. *Tools for thinking: modelling in management science*, Wiley, New York.

Pillana, S. and T. Fahringer, T. 2002. UML based modelling of performance oriented parallel and distributed applications. In *Proceedings of the 2002 Winter Simulation Conference*, ed. J. L. Snowdon, J. M. Charnes, E. Yucesan, and C. Chen. 497-505. San Diego, CA, USA.

Pooch, U. W. and J. A. Wall. 1993. *Discrete Event Simulation: A Practical Approach*, CRC Press Inc., London.

Pooley, R. J. and P. H. Hughes. 1991. 'Towards a standard for hierarchical process oriented discrete event simulation diagrams: The suggested approach to flat models. *Transactions of the Society for Computer Simulation*, vol. 8, no. 1, 21-31.

Pratt, D. B., M. Kamath, and J. H. Mize. 1995. A comprehensive Modelling and analysis environment for manufacturing systems. In *Proceedings of the 4th Industrial Engineering Research Conference*, Nashville, TN, 759-768. Institute of Industrial Engineers.

- Pratt, D. B., P. A. Farrington, C. B. Basnet, H. C. Bhuskute, M. Kamath, and M. and J. H. Mize. 1994. The separation of physical, information and control elements for facilitating reusability in simulation modelling. *International Journal of Computer Simulation*. Vol. 4, 327-342.
- Randell, L.G., L.G. Holst, and G.S. Bolmsjo. 1999. Incremental system development of large discrete-event simulation models. In *Proceedings of the Winter Simulation Conference*, ed. D. T. Sturrock, G. W. Evans, P. A. Farrington, and H. B. Nemhard. 561-568. Phoenix, AZ, USA.
- Rathmell, J. and D. T. Sturrock. 2002. Arena product family: enterprise modelling solutions. In *Proceedings of the 2002 Winter Simulation Conference*, ed. J. L. Snowdon, J. M. Charnes, E. Yucesan, and C. Chen. 165-172. San Diego, CA , USA.
- Richter, H. and L. Marz. 2000. Towards a standard process: the use of UML for designing simulation model. In *Proceedings of the 2000 Winter Simulation Conference*, ed. P. A. Fishwick, K. Kang, J. A. Joines, and R. H. Barton. 394-398. Orlando, FL, USA.
- Rio, I. D. 1999. Comment made on “inquiry on model representation and documentation”, *Abstraction Software Company*.
- Robinson, S. 1994. *Successful Simulation: Practical Approach to Simulation*, McGraw-Hill Inc., Berkshire, UK.
- Rossetti, M., B. Aylor, R. Jacoby, A. Prorock, and A. White. 2000. SIMFONE: an object-oriented simulation framework. In *Proceedings of the 2000 Winter Simulation Conference*, ed. P. A. Fishwick, K. Kang, J. A. Joines, and R. H. Barton. 1855-1864. Orlando, FL, USA.
- Russo, C. W. R. 1997. 12 rules to make your ISO 9000 documentation simple and easy to use. *Quality Progress*, March, 51-53.
- Sadoun, B. 2000. Applied system simulation: a review study. *Information Science*, vol. 124, 173-192.
- Sadowski. D.A. and M.R. Gradau. 2000. Tips for successful practice of simulation. In *Proceedings of the 2000 Winter Simulation Conference*, ed. P. A. Fishwick, K. Kang, J. A. Joines, and R. R. Barton. 26-31. Orlando, FL, USA.
- Sargent, R. G. 2000. Verification, validation, and accreditation of simulation models. In *Proceedings of the 2000 Winter Simulation Conference*, ed. P. A. Fishwick, K. Kang, J. A. Joines, and R. H. Barton. 50-59. Orlando, FL, USA.
- Saunders, M., P. Lewis and A. Thornhill, 2003. *Research Methods for Business Students*, 3rd edn., Pearson Educational Ltd., Essex.
- Sawhney, A., O. Abudayyeh, and A. Monga. 1999. Modelling and analysis of a mail processing plant using Petri Nets. *Advances in Engineering Software*, vol. 30, 543-549.
- Schormann, F. and T. Perera. 1999. *Model representation methods in simulation of manufacturing systems*. An unpublished paper. Sheffield Hallam University, Sheffield.
- Seppanen M. S. 2000. Developing industrial strength simulation models using Visual Basic for applications. In *Proceedings of the 2000 Winter Simulation Conference*, ed. P. A. Fishwick, K. Kang, J. A. Joines, and R. H. Barton. 72-82. Orlando, FL, USA.
- Shannon, R. E. 1992. Introduction to simulation. In *Proceedings of the 1992 Winter Simulation Conference*, ed. R. C. Crain, J. R. Wilson, J. J. Swain, and D. Glodsmen. 65-73. Arlington, VA, USA.

- Shannon, R. E. 1998. Introduction to simulation. In *Proceedings of the 1998 Winter Simulation Conference*, ed. J. S. Carson, M. S. Manivannan, D. J. Medeiros, and E. F. Watson. 7-14. Washington DC, USA.
- Shi, J. J. 2000. Object-oriented technology for enhancing activity-based modelling functionality: an example. In *Proceedings of the 2000 Winter Simulation Conference*, ed. P. A. Fishwick, K. Kang, J. A. Joines, and R. H. Barton. 1398-1444. Orlando, FL, USA.
- Simule, 2001. AutoMod simulation software. <http://www.autosim.com>. [16th April 2002]
- Son, Y. J., A. T. Jones and R. A. Wysk. 2000. Automatic generation of simulation models from neutral libraries. In *Proceedings of the 2000 Winter Simulation Conference*, ed. P. A. Fishwick, K. Kang, J. A. Joines, and R. H. Barton. 1558-1567. Orlando, FL, USA.
- Spear, B. 1984. *How to Document Your Software*, Tabs Books Inc., USA.
- Stahl, I. 2002. Simulation prototyping. In *Proceedings of the 2002 Winter Simulation Conference*, ed. J. L. Snowdon, J. M. Charnes, E. Yucesan, and C. Chen. 572-579. San Diego, CA, USA.
- Takus, D. A. and D. M. Profozich. 1997. Arena software tutorial. In *Proceeding of the 1997 Winter Simulation Conference*, ed. D. H. Withers, B. L. Nelson, S. Andradottir, and K. J. Healy. 541-544. Atlanta, GA, USA.
- Taylor, S. J. and R. Bogdan, 1984. *Introduction to Qualitative Research Methods*, John Wiley & Sons.
- Tesham, B. and E. Unger. 1996. A simulation model for evaluating personnel schedules in hospital emergency department. In *Proceedings of the 1996 Winter Simulation Conference*, ed. J. M. Charnes, D. J. Morrice, D. T. Brunner, and J. J. Swain. 1205-1209. Coronado, CA, USA.
- Thomas, W. M. 1997. An analysis of errors in a reuse-oriented development environment', *Journal of System Software*, vol. 38, 211-224.
- Tye, B. S. 1999. *A study on model design in the simulation of manufacturing systems*. An unpublished PhD thesis. Sheffield Hallam University, Sheffield.
- Vliet, H. V. 2000. *Software engineering principles and practice*, John Wiley & Sons, Ltd.
- Vojnar, T. 1997. Various kinds of Petri Nets in simulation and modelling. *Proceedings of the 31st Spring International Conference on Modeling and Simulation of Systems*, MOSIM97.
- Watson, H. J. and J. H. Blackstone. 1989. *Computer simulation*, 2nd edn., John Wiley & Sons, Inc.
- Whitman, L., B. Huff, and A. Presley. 1997. Structured models and dynamic system analysis: the integration of the IDEF0/IDEF3 modelling methods and discrete event simulation. In *Proceeding of the 1997 Winter Simulation Conference*, ed. D. H. Withers, B. L. Nelson, S. Andradottir, and K. J. Healy. 518-524. Atlanta, GA, USA.
- William, E. J., O. M. Ulgen, and C. De Witt. 2002. An approach and interface for building generic manufacturing Kanban systems models. In *Proceedings of the 2002 Winter Simulation Conference*, ed. J. L. Snowdon, J. M. Charnes, E. Yucesan, and C. Chen. 1138-1141. San Diego, CA, USA.
- Yin, R. E., 1994. *Case study research: design and methods*, Saga Publications.

ANNEXURE “A” – MAPPING TABLES

Documentation Model Database			ARENA Executable Model Database			
Element/ Table	Field Description	Field Name	Prompt	Field Name	Field Description	Table Name
PE/ Entity	Name for model level	P_ModelLevel Name	S Auto			
	Unique ID for a Master drop	ShapeKey	Auto	SerialNumber	Unique ID for a module	BasicProcess Entity
	ID for model level	P_ModelLevelID	S Auto	ModelLevelID	ID for model level	
	X – Element location	PinX	Auto	X (= 0)	X – Module location	
	Y – Element location	PinY	Auto	Y (= 0)	Y – Module location	
	Unique name of item arriving	P_ShapeName	Input	Name	Unique entity type	
	Name for model level	P_ModelLevel Name	S Auto			
PE/ Processor	Unique ID for a Master drop	ShapeKey	Auto	SerialNumber	Unique ID for a module	BasicProcess Resource
	ID for model level	P_ModelLevelID	S Auto	ModelLevelID	ID for model level	
	X – Element location	PinX	Auto	X (= 0)	X – Module location	
	Y – Element location	PinY	Auto	Y (= 0)	Y – Module location	
	Unique name for processor	P_ShapeName	Input	Name	Unique resource name	
	Processor capacity based on	P_CapacityBasedOn (=Schedule)	Auto	Type	Capacity type (fixed/sched)	
	ID for pro cycle (sch & failure)	P_ResCycleID (=Rcy_ShapeName)	Auto	Schedule	Capacity by schedule name	
	Format or Schedule or failure rule	P_Rule (=Wait/Preempt/Ignore)	Auto	Schedule Rule	Rule for schedule	
	No. of processor units	P_NoOfUnits	Input	Quantity	Resource quantities	
	Process name(s) for documentation	P_OperationNames (process,...)	Input			
	Format of processing time units	P_TimeUnits(=Days/Hours/Minutes)	Auto			
	Attached station name	P_StationName (=Sta_ShapeName)	Auto/In			
	Format for schedule time per day	P_TimeUnitsPerDay (=1,60,True,10,0,1,1)	Auto			
	Format of schedule values	P_RCyScheduleValues (=Cap*Dur,...)	Auto			
	Failure type depends on	P_FailureType (=Time)	Auto			
	Format of failure values	P_RCyFailureValues (=Con/Ex(mean)..)	Auto			
	Name for model level	P_ModelLevel Name	S Auto			
PE/ Output	Unique ID for a Master drop	ShapeKey	Auto			
	ID for model level	P_ModelLevelID	S Auto			
	X – Element location	PinX	Auto			
	Y – Element location	PinY	Auto			
	Unique name for item departing	P_ShapeName	Input			

Equivalent Tables in Model Documentation (VISIO) and Implementation (ARENA) Model Databases

Documentation Model Database				ARENA Executable Model Database			
Element/ Table	Field Description	Field Name	Prompt	Prompt	Field Name	Field Description	Table Name
PE/ Storage	Name for model level	P_ModelLevel Name	SAuto	SAuto			
	Unique ID for a Master drop	ShapeKey	Auto	Auto	SerialNumber	Unique ID for a module	Advanced Process Resource
	ID for model level	P_ModelLevelID	SAuto	SAuto	ModelLevelID	ID for model level	
	X – Element location	PinX	Auto	Auto	X (= 0)	X – Module location	
	Y – Element location	PinY	Auto	Auto	Y (= 0)	Y – Module location	
	Unique name for storage	P_ShapeName	Input	Input	Name	Unique storage name	
	No. of storage units	P_NoOfUnits	Input	Input			
	Store/unstore name(s) for document	P_OperationNames {store,unstore,...}	Input	Input			
	Attached station name	P_StationName (=Sta_ShapeName)	Auto/In	Auto/In			
	Storage capacity based on	P_CapacityBasedOn (=Schedule)	Auto	Auto			
	ID for storage cycle (sch & failure)	P_ResCycleID (=Rcy_ShapeName)	Auto	Auto			
	Format or Schedule or failure rule	P_Rule (=Wait/Preempt/Ignore)	Auto	Auto			
	Format for schedule time per day	P_TimeUnitsPerDay (=1,60,True,10,0,1,1)	Auto	Auto			
	Format of schedule values	P_RCYScheduleValues (=Cap*Dur...)	Auto	Auto			
	Failure type depends on	P_FailureType (=Time)	Auto	Auto			
PE/ Transporter	Format of failure values	P_RCYFailureValues (=Con/Ex(mean...))	Auto	Auto			
	Name for model level	P_ModelLevel Name	SAuto	SAuto			
	Unique ID for a Master drop	ShapeKey	Auto	Auto	SerialNumber	Unique ID for a module	Advanced Process Resource
	ID for model level	P_ModelLevelID	SAuto	SAuto	ModelLevelID	ID for model level	
	X – Element location	PinX	Auto	Auto	X (= 0)	X – Module location	
	Y – Element location	PinY	Auto	Auto	Y (= 0)	Y – Module location	
	Unique name for transporter	P_ShapeName	Input	Input	Name	Unique transporter name	
	Transporter docked station(s)	P_DockedStation	Input	Input	Name	Docked position	
	No. of transporter units	P_NoOfUnits	Input	Input			
	Transport name(s) for documentation	P_OperationNames (transport,...)	Input	Input			
	Attached station name	P_StationName (=Sta_ShapeName)	Auto/In	Auto/In			
	ID for transporter velocity	P_Velocity (=Velo_ShapeName)	Auto	Auto			
	Format for transporter velocity units	P_VeloUnits (m/s,cm/s,min/s)	Auto	Auto			
	Transporter capacity based on	P_CapacityBasedOn (=Schedule)	Auto	Auto			
	ID for transporter cycle (sch&failure)	P_ResCycleID (=Rcy_ShapeName)	Auto	Auto			
	Format or Schedule or failure rule	P_Rule (=Wait/Preempt/Ignore)	Auto	Auto			
	Format for schedule time per day	P_TimeUnitsPerDay (=1,60,True,10,0,1,1)	Auto	Auto			

Equivalent Tables in Model Documentation (VISIO) and Implementation (ARENA) Model Databases

Documentation Model Database				ARENA Executable Model Database			
Element/ Table	Field Description	Field Name	Prompt	Prompt	Field Name	Field Description	Table Name
PE/ Transporter (Contd..)	Format of schedule values	P_RCyScheduleValues (=Cap*Dur...)	Auto				
	Failure type depends on	P_FailureType (=Time)	Auto				
	Format of failure values	P_RCyFailureValues (=Con/Ex(mean)..)	Auto				
PE/ Conveyor	Name for model level	P_ModelLevel Name	SAuto				
	Unique ID for a Master drop	ShapeKey	Auto	Auto	SerialNumber	Unique ID for a module	Advanced Process Resource
	ID for model level	P_ModelLevelID	SAuto	Auto	ModelLevelID	ID for model level	
	X – Element location	PinX	Auto	Auto	X (= 0)	X – Module location	
	Y – Element location	PinY	Auto	Auto	Y (= 0)	Y – Module location	
	Unique name for Conveyor	P_ShapeName	Input	Input	Name	Unique conveyor name	
	Transporter docked station(s)	P_DockedStation	Input				
	No. of conveyor units	P_NoOfUnits	Input				
	Convey name(s) for documentation	P_OperationNames (convey,...)	Input				
	Attached station name	P_StationName (=Sta_ShapeName)	Auto/In				
	ID for convey velocity	P_Velocity (=Velo_ShapeName)	Auto				
	Format for convey velocity units	P_VeloUnits (m/s, cm/s, mm/s)	Auto				
	Conveyor capacity based on	P_CapacityBasedOn (=Schedule)	Auto				
	ID for transporter cycle (sch&failure)	P_ResCycleID (=Rcy_ShapeName)	Auto				
IE/ Departure	Format or Schedule or failure rule	P_Rule (=Wait/Preempt/Ignore)	Auto				
	Format for schedule time per day	P_TimeUnitsPerDay (=1,60,True,10,0,1,1)	Auto				
	Format of schedule values	P_RCyScheduleValues (=Cap*Dur...)	Auto				
	Failure type depends on	P_FailureType (=Time)	Auto				
	Format of failure values	P_RCyFailureValues (=Con/Ex(mean)..)	Auto				
	Name for model level	P_ModelLevel Name	SAuto				
	Unique ID for a Master drop	ShapeKey	Auto	Auto	SerialNumber	Unique ID for a Module	Basic Process Dispose
	ID for model level	P_ModelLevelID	SAuto	Auto	ModelLevelID	ID for model level	
	X – Element location	PinX	Auto	Auto	X	X – Module location	
	Y – Element location	PinY	Auto	Auto	Y	Y – Module location	
	Unique name for item departing	P_ShapeName	Input	Input	Name	Unique dispose name	
	Name of item departing	P_ItemNameDeparts	Input				

Equivalent Tables in Model Documentation (VISIO) and Implementation (ARENA) Model Databases

Documentation Model Database				ARENA Executable Model Database			
Element/ Table	Field Description	Field Name	Prompt	Prompt	Field Name	Field Description	Table Name
IE/ Arrival	Name for model level	P_ModelLevel Name	SAuto				
	Unique ID for a Master drop	ShapeKey	Auto		SerialNumber	Unique ID for a module	Basic Process Create
	ID for model level	P_ModelLevelID	SAuto		ModelLevelID	ID for model level	
	X – Element location	PinX	Auto		X	X – Module location	
	Y – Element location	PinY	Auto		Y	Y – Module location	
	Unique name for entity arrival	P_ShapeName	Input		Name	Unique create name	
	Name of the arriving entity	P_ArrivalEntityName	Input		Entity Type	Entity name created	
	No. of entities per arrival	P_NoOfEntitiesPerArrival	Input		Batch Size	No. of entities per create	
	Arrival time behaviour	P_TimeBehaviour (const/expr)	Select		Inter-arrival Type	Mode of inter-arrival time	
	Time between two arrivals	P_TimeBetweenArrivals	Input		Inter-arrival Type	Inter-arrival type	
	Time units used inter-arrival	P_TimeUnits	Select		Units	Inter-arrival time units	
	Name for model level	P_ModelLevel Name	SAuto				
CE/ Priority	Unique ID for a Master drop	ShapeKey	Auto				
	ID for model level	P_ModelLevelID	SAuto				
	X – Element location	PinX	Auto				
	Y – Element location	PinY	Auto				
	Unique name for priority identity	P_ShapeName	Input				
	Priority for operation (OR)	P_OperationsPriority(OperName-1/2/3,...)	Input		Priority value(H/M/L)	Priority value selected	BP Process
	Priority for capture (OR)	P_CapturePriority(CapName-1/2/3,...)	Input		Priority value(H/M/L)	Priority value selected	AP Seize
	Name for model level	P_ModelLevel Name	SAuto				
	Unique ID for a Master drop	ShapeKey	Auto		SerialNumber	Unique ID for a Module	Basic Process Process
	ID for model level	P_ModelLevelID	SAuto		ModelLevelID	ID for model level	
	X – Element location	PinX	Auto		X	X – Module location	
	Y – Element location	PinY	Auto		Y	Y – Module location	
	Unique name for process	P_ShapeName	Input		Name	Unique process name	
	ID for process time in expression	P_ProcessDelayID(=Time_P_ShapeName)	Input		Expression	Name of expression	
	Format of process time units	P_TimeUnits(=Days/Hours/Minutes)	Auto		Units	Time units in process	
	Category of the process	P_OperationCategory (≠D or SDR)	AuSel		Action	Actions in the process	
	Process time expressed by	P_TimeBehaviour (≠Expression)	Auto		DelayType	Time by expression	
	Type: a set OR resource(s)	P_IndexType (a set/resource)	Select		ResourceType	Type of the resource	
IE/ Process OR CE/Delay	Set/Processor name(s) in the process	P_IndexNames (set/processor,...)	Input				BP Process Resource
	No. of Entities per Unit Processor	P_NoOfEntitiesPerUnitRes (=1)	Input				

Equivalent Tables in Model Documentation (VISIO) and Implementation (ARENA) Model Databases

Documentation Model Database				ARENA Executable Model Database		
Element/ Table	Field Description	Field Name	Prompt	Field Name	Field Description	Table Name
LE/ Capture	Name for model level	P_ModelLevel Name	SAuto			
	Unique ID for a Master drop	ShapeKey	Auto	SerialNumber	Unique ID for a Module	Advanced Process
	ID for model level	P_ModelLevelID	SAuto	ModelLevelID	ID for model level	Seize
	X – Element location	PinX	Auto	X	X – Module location	
	Y – Element location	PinY	Auto	Y	Y – Module location	
	Unique name for capture	P_ShapeName	Input	Name	Unique seize name	
	Queue for capture	P_CaptureQueueName(Q_P.shapename)	Auto	QName	Queue name for capture	
	Type: a set OR resource(s)	P_IndexType (a set/resource)	Select	ResourceType	Type of the resource	AP Seize Resource
	Set/Processor name(s) in the capture	P_IndexNames (set/processor1,...)	Input			
	Name for model level	P_ModelLevel Name	SAuto			
LE/Free	Unique ID for a Master drop	ShapeKey	Auto	SerialNumber	Unique ID for a Module	Advanced Process
	ID for model level	P_ModelLevelID	SAuto	ModelLevelID	ID for model level	Release
	X – Element location	PinX	Auto	X	X – Module location	
	Y – Element location	PinY	Auto	Y	Y – Module location	
	Unique name for free	P_ShapeName	Input	Name	Unique release name	
	Type: a set OR resource(s)	P_IndexType (a set/resource)	Select	ResourceType	Type of the resource	AP Release Resource
	Set/Processor name(s) in the free	P_IndexNames (set/processor1,...)	Input			
	Name for model level	P_ModelLevel Name	SAuto			
	Unique ID for a Master drop	ShapeKey	Auto	SerialNumber	Unique ID for a Module	BasicProcess
	ID for model level	P_ModelLevelID	SAuto	ModelLevelID	ID for model level	Set
IE(Auxiliary Ele)/Set	X – Element location	PinX	Auto	X (=0)	X – Module location	OR
	Y – Element location	PinY	Auto	Y (=0)	Y – Module location	AdvancedProcess Adv
	Unique name for set	P_ShapeName	Input	Name	Unique release name	anced Set
	Pr/St/Tr/Con/Sta type in the set	P_IndexType (Pr/St/Tr/Con/Sta)	Select	Type	Type of the set	
	Members in the set	P_IndexNames (Member1,...)	Input	Resource/... name	Members of the set	BP Set Members OR
						AP Aset Members

Equivalent Tables in Model Documentation (VISIO) and Implementation (ARENA) Model Databases

Documentation Model Database				ARENA Executable Model Database			
Element/ Table	Field Description	Field Name	Prompt	Prompt	Field Name	Field Description	Table Name
CE/ Selection OR LE/ Selection	Name for model level	P_ModelLevel Name	SAuto				
	Unique ID for a Master drop	ShapeKey	Auto				
	ID for model level	P_ModelLevelID	SAuto				
	X – Element location	PinX	Auto				
	Y – Element location	PinY	Auto				
	Unique name for selection	P_ShapeName(=Sel_P_Setnameforddecision)	Auto				
	Set name associated in selection	P_SetNameForDecision	Input	Input	Set name for select.	Set name	BP Process Resources
	Defined/undefined selection rule	P_Rule	Sel/In	Select	Selection Rule	Rule for set member selection	BP Process Resources
	Att/Var name associated in rule,if	P_AttributeVariableName	Input				
	Operation name in selection (OR)	P_OperationNameForDecision	Input				
CE/ Distribution (-Out)	Capture name in selection (OR)	P_CaptureNameForDecision	Input				
	Name for model level	P_ModelLevel Name	SAuto				
	Unique ID for a Master drop	ShapeKey	Auto	Auto	SerialNumber	Unique ID for a Module	BasicProcess
	ID for model level	P_ModelLevelID	SAuto	Auto	ModelLevelID	ID for model level	Decide
	X – Element location	PinX	Auto	Auto	X (=0)	X – Module location	
	Y – Element location	PinY	Auto	Auto	Y (=0)	Y – Module location	
	Unique name for distribution	P_ShapeName	Input	Input	Name	Unique decision name	
	Type of distribution	P_Type (With/NWWith)	Auto	Select	Type	Decision type	
	Values in distribution	P_IndexNames (Value1,...)	Input	Input	Percent True OR N Percent True	True values in choices	BP Decide OR BP Decide Conditions

Equivalent Tables in Model Documentation (VISIO) and Implementation (ARENA) Model Databases

NOTE:	PE	Physical Element	CE	Control Element	DE	Data Element
	PIE	Primary Information Element	AIE	Auxiliary Information Element	P_	Property in Visio Data Sheet
Display only a few numbers of elements for demonstration purpose and the feasibility was studied only for elements related to process system (excluding transport and convey).						

ANNEXURE “B” – QUESTIONNAIRE SURVEY: VALIDATION OF THE PROPOSED METHODOLOGY

A Survey to Validate the Proposed Methodology for Model Representation and Documentation (MRD) in Computer Simulation

For the purpose of encouraging (rather helping) modellers to document simulation projects, a documentation system [referred as MRD system] was developed. It is a structured methodology based on ‘task-orientation’ and enriched with ‘reusable model elements’. It is also an integrated system that enables the project team to document a simulation project concurrently with model development process benefiting for both processes mutually. The proposed MRD system is expected to be generic and applicable for documentation of any simulation project, regardless of the simulation software being used. It is also expected that the MRD system should not only guide the project team to record project details enabling to provide the necessary information for their day to day management and operational tasks and for model reuse but also direct them to perform a simulation project step by step with right path to ensure that the project is successful.

Therefore, the aim of this questionnaire is primarily to obtain the views from experts in order to evaluate and validate the proposed methodology. More specifically, this aims;

- To obtain the characteristics of current practices in documentation of simulation project,
- To attest the characteristics to be embedded in a documentation system,
- To confirm the applicability of task-oriented approach to document the details of a simulation project,
- To substantiate that the proposed methodology is generic and standardized, and
- To validate that the proposed MRD system is practical and viable.

An attempt is made to present the questionnaire self-explanatory. But some of new terms are explained in ‘Supplementary Details’ file and they are underlined and hyper-linked in appropriate questions.

1. About You:

1.1 Which of following best describes your current role in respect of simulation?

- ☐ Academic ☐ Company (internal) ☐ Consultant (external) ☐ Researcher
☐ Other (Please specify):

1.2 Roughly, how many simulation models in real world applications have you built so far?

- ☐ None ☐ 1 to 5 ☐ 6 to 10
☐ 11 to 15 ☐ 16 to 20 ☐ > 20

1.3 Please state which simulation software that you have more experience with (e.g. Arena, Quest).

Simulation Software:

2. Your Practice and General Opinion about Documentation in Simulation:

2.1 Do you generally document the details of simulation project?

- ☐ Always ‘Yes’ ☐ Usually ‘Yes’ ☐ Sometimes ‘Yes’
☐ Seldom ‘Yes’ ☐ Always ‘No’ ☐ No experience in real application

2.2 If ‘Yes’ for 2.1, please answer the followings to describe how you perform the documentation process.

A. What level of details do you generally document in a simulation project? (Please select one).

- ☐ Minimal details like conclusions, recommendations, etc. to fulfil the needs of the client
☐ Moderate details to assist in the modelling process, model run and model implementation
☐ Comprehensive details to benefit for the all team members, model re-users and others interested
☐ Other (Please specify):

B. How do you generally perform documentation process? (Please select appropriately and specify).

- ☐ Retrospectively (after project is completed) ☐ Progressively (as the project progresses)
☐ Both progressively and retrospectively
☐ Always independently from the software ☐ Usually independently from the software
☐ Mostly depend on the software
☐ Other (Please specify):

C. What are the factor(s) that you were obstructed in the documentation process?

- ☐ Difficulty in identification of significant details for documentation
- ☐ Non-availability of standard structured methodology and guide-lines for documentation
- ☐ Treating documentation as isolated process from model development
- ☐ Lack of time frame for proper documentation
- ☐ Tendency of forgetting the details, in case of retrospective documentation
- ☐ Other (Please specify):

2.3 If 'No' for 2.1, please answer the following to describe why you do not document simulation projects.

A. What are the factor(s) that you were made to avoid the documentation of simulation project?

- ☐ Non-availability of time for documentation within the time frame allocated
- ☐ Non-availability of standard structured methodology and guide lines for documentation
- ☐ Other (Please specify):

2.4 Please cross 'X' in appropriate box on how you agree with the statements presented about the characteristics to be incorporated in a documentation process of simulation project.

SA-Strongly Agree; A-Agree; DA-Disagree; SDA- Strongly Disagree; NA/DA-Neither Agree or Disagree		SA	A	DA	SDA	NA/DA
A.	Documentation should be progressive, allowing to record details more effectively than retrospective documentation.					
B.	Documentation should be independent from the simulator and be in generic terms to enhance the understanding.					
C.	Documentation system should have a common layout with structured and linked details.					
D.	Documentation should be integrated with model development to benefit for both processes mutually.					

3. Model Representation and Documentation (MRD) Process:

Documentation in simulation has been named as Model Representation and Documentation (MRD) to avoid varying interpretation found in literature and to accommodate the details of the entire project and the significance of the model representation task. The proposed 'Definition of MRD' aims to fulfil various needs of different audiences in respect of the current project, model reuse and others interested like researchers.

3.1 Do you think that such an explicit definition is necessary for the documentation of simulation projects?

- ☐ Definitely 'Yes' ☐ Yes ☐ No ☐ Definitely 'No' ☐ Not Sure

3.2 Do you feel that the definition needs to be expanded further to accommodate any missing purposes?

- ☐ Yes ☐ No ☐ Not Applicable

If 'Yes', please state any other details you suggest:

We have proposed 'Task-Orientation' (more specifically task-documentation) approach for MRD. It offers the user to manage the documentation process with micro level of task documents progressively with model development process and accumulate them to produce documents to fulfil various needs of different audience.

3.3 Do you agree that the task-documentation is the most appropriate approach for documentation of simulation projects for fulfilling the purposes stated in the 'Definition of MRD'?

- ☐ Strongly Agree ☐ Agree ☐ Disagree
- ☐ Strongly Disagree ☐ Neither Agree or Disagree

If 'Disagree', please state why you disagree or and what alternatives you propose:

..... <Please go to Other Remarks>

'Task Documents' were identified by breaking the project procedure (we have proposed a revised procedure) into micro level of tasks. Pre-structured forms (e.g. tables) for almost all task documents were then suggested to record the details of any project.

3.4 Do you think that the use of standard project life cycle (or revised procedure) represent a generic framework for task documents enabling to apply for documentation of any type of simulation project?

[] Definitely 'Yes' [] Yes [] No [] Definitely 'No' [] Not Sure

If 'No', please state why do you think that it doesn't represent:

..... <Please go to Other Remarks>

3.5 Please 'X' in appropriate box on how you agree with the statements presented on benefits expected from task documents and their pre-structured forms for documentation and model development processes.

SA-Strongly Agree; A-Agree; DA-Disagree; SDA- Strongly Disagree; NA/DA-Neither Agree or Disagree		SA	A	DA	SDA	NA/DA
A.	Task documents serve as a checklist not only for documentation but also for model development process.					
B.	Pre-structured forms in task documents provide a uniform structure for presentation of task details.					
C.	Pre-structured forms in task documents encourage/help the modellers for documentation.					
D.	Pre-structured forms guide the user to perform tasks with right direction without misleading the process.					
E.	Pre-structured forms lay a sound foundation for data exchange and for integrated documentation system.					

Software independent task details were addressed with the use of 'Reusable Model Elements' that collectively represent the behaviour and the logics of the model graphically and capture model data for documentation purpose. In identification of task details, emphasize was given on these reusable model elements in each document, wherever possible, in keeping the 'Documentation Model' in focus.

3.6 Do you think that use of reusable model elements represents software independent task details for documentation?

[] Definitely 'Yes' [] Yes [] No [] Definitely 'No' [] Not Sure

3.7 Do you think that use of reusable model elements helps the user to describe the model explicitly?

[] Definitely 'Yes' [] Yes [] No [] Definitely 'No' [] Not Sure

3.8 Do you think that amalgamated reusable model elements and flowchart symbols allow the user to describe the model software independently?

[] Definitely 'Yes' [] Yes [] No [] Definitely 'No' [] Not Sure

4. MRD System:

MRD system is defined as a documentation management and control system that has a common layout and systematically linked and interfaced details in each task of the project. It is expected that the proposed MRD system should advise and encourage/help the project team not only to document the project but also to perform the project with reasonable endeavour benefiting from documentation.

4.1 A common or standard layout for documentation is maintained with documentation templates that comprise pre-structured forms defined for task documents so that the user can use the templates to record the details of task documents for different projects.

A. Do you think that use of common templates could encourages/helps project team for documentation?

[] Definitely 'Yes' [] Yes [] No [] Definitely 'No' [] Not Sure

If 'No', please state why:

4.2 Some data may be required to repeat in different level of task documents in order to maintain the modularity of task documents to serve different audience. These details are shared by linking the details of task documents.

A. Do you think that linked details of task documents encourage/help project team for documentation?

[] Definitely 'Yes' [] Yes [] No [] Definitely 'No' [] Not Sure

If 'No', please state why:

4.3 What are capabilities that you think to be incorporated for handling task documents in MRD system (or tool) that advises and encourages/helps project team to document the project? [Please 'X' your choice(s)].

- ☐ A. It should display the different stages of the simulation project and task documents clearly.
- ☐ B. It should make any task documents accessible at any point in time to accommodate sequential or iterative modelling process.
- ☐ C. It should visualize the contents of task documents immediately.
- ☐ D. It should allow the alteration to task details with least interaction.
- ☐ E. It should distinguish between the task documents recorded and not recorded.
- ☐ F. It should indicate previous task documents that require attention or modification in case of an alteration made to a particular task document.
- ☐ G. Other (Please specify):

For handling task documents, we have incorporated 'Tree-View, Immediate Display Screen, Check-Boxes and Highlights of Task Documents' features to the MRD system (or MRD Tool) to accommodate the capabilities (A&B), (C&D), (E) and (F) respectively.

4.4 Please suggest any other features that you think are useful to be incorporated for handling task documents in MRD system (or MRD tool) to accommodate the capabilities suggested or proposed by you.

Features for handling task documents:

4.5 What are the other facilities that you think to be incorporated in MRD system (or MRD tool) that may be useful for an efficient documentation system? [Please 'X' your choice(s)]

- ☐ P. It should be capable of communicating details among task documents that may be in the same software (e.g. MS Word) or different software (e.g. Word and Visio) for data sharing.
- ☐ Q. It should facilitate to produce uniform and consistent documents for different purposes and different audiences in any stage of the project.
- ☐ R. Other (Please specify):

For an efficient documentation system, we have incorporated 'Interfaces for Data Sharing' and 'Interfaces for Document Preparation' to the MRD system (or MRD Tool) to accommodate (P) and (Q) respectively.

4.6 Please suggest any other interfaces that you think useful to be incorporated to MRD system (or MRD tool) to accommodate the facilities suggested or proposed by you.

Interfaces for an efficient documentation system:

5. Integrated MRD System:

Even though the MRD System may be successful for documentation, it is still a process isolated from model development, running as two distinct processes. Therefore, they were integrated to execute them as a single process in order to be benefited for both processes mutually. It is expected that such an integrated MRD system inevitably improves the documentation further and the modellers may then treat the documentation as an integral activity in a simulation project.

Integrated MRD system is basically facilitated by model exchange, i.e. translating 'Documentation Model' to software dependent executable model and vice-a-versa, through a 'Common Database'.

5.1 The feasibility of such a common database was studied with Arena software for process related model. And it has been proved that translation of 'Documentation Model' to executable model can be performed with the same data utilized for documentation purpose.

A. Do you think that integrated MRD system further encourages/helps modellers for documentation?

☐ Definitely 'Yes' ☐ Yes ☐ No ☐ Definitely 'No' ☐ Not Sure

If 'No', please state why:

B. Do you think that use of a ‘Common Database’, which stores the information of the model, lays a foundation for model interoperability and model reusability?

[] Definitely ‘Yes’ [] Yes [] No [] Definitely ‘No’ [] Not Sure

If ‘No’, please state why:

C. Do you think that use of proposed common database, which comprised with details of reusable model elements that are stored with standardized internal structure, makes provision to translate the documentation model into executable in any simulation software?

[] Definitely ‘Yes’ [] Yes [] No [] Definitely ‘No’ [] Not Sure

If ‘No’, please state why:

5.2 With the development of ‘Standard MRD System’, it is expected that MRD system could be implemented as stand alone commercial software interacting with diverse nature of simulation software for model exchange process, or else, as a separate system module interfaced to individual software.

A. Do you think that such a ‘Standard MRD System’ is viable and practicable?

[] Definitely ‘Yes’ [] Yes [] No [] Definitely ‘No’ [] Not Sure

If ‘No’, please state why:

6. Overall Assessment:

6.1 As far as I understand the methodology presented in the questionnaire, I think; (please ‘X’ in appropriate box on how you agree with statements presented in respect of the proposed methodology).

SA–Strongly Agree; A–Agree; DA–Disagree; SDA– Strongly Disagree; NA/DA–Neither Agree or Disagree		SA	A	DA	SDA	NA/DA
A.	The MRD process is time consuming than I expected.					
B.	The MRD process is lengthier than I expected.					
C.	The MRD process is very useful for the success of simulation project and for enhancing model reuse.					
D.	The MRD system encourages/helps the modeller to document simulation project.					
E.	The MRD system acts as management, control and monitoring system of simulation project.					
F.	The MRD system makes the documentation to treat as an integral activity in a simulation project.					
G.	The MRD system acts as a training system to a novice user in simulation.					
H.	The documentation model with common database acts as a media for model interoperability & enhances reusability					

6.2 According to my experience with (please specify) simulation software, about the translation of documentation model, which are claimed to be software independent and exchangeable, to executable model **would be highly-likely/likely/unlikely/highly-unlikely** or **cannot be commented** (please cross over inappropriate choices or delete).

7. Other Remarks:

Please state any other positive or negative remarks that you think would be useful in respect to the proposed methodology or any other comments and suggestions in respect of documentation in simulation project.

.....

***** Thank You for Your Valuable Time and Comments *****

SUPPLEMENTARY DETAILS

Definition of MRD

MRD is defined as recording the knowledge, experience and results embedded with each step in a simulation project to fulfil the following purposes. Fig. 1 illustrates how different audience benefit from the proposed notion.

- **Communication:** To communicate the state of the simulation project.
- **Dissemination:** To disseminate the accumulated knowledge and experience.
- **Motivation:** To keep the enthusiasm of the team high.
- **Quality:** To ensure the quality and creditability of the model and the project.
- **Reporting:** To report the results and findings of the project.
- **Maintenance:** To guarantee the project progress or error traceable.
- **Contingency:** To protect against time and knowledge losses (personnel turnover).
- **Enhancement:** To enhance the effective model reuse.

<div><div></div><div>Audience Benefit from Documentation</div></div>	Model User (Owner)								Model Re-User		Others Interested	
	The Client	Team/Project Leader	The Modeler	System Supporter	Modeling Supporter	Data Provider	Model Executor	Interested In-house Member	Client/Top-Level Manager	Analysr/Modeler	Researcher/Student	Others
Purpose of Documentation												
Dissemination	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Quality	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Maintenance	✓	✓	✓	✓	✓	✓	✓	✓	x	x	x	x
Communication	✓	✓	✓	✓	✓	✓	✓	✓	x	x	x	x
Motivation	✓	✓	✓	✓	✓	✓	✓	x	x	x	x	x
Reporting	✓	✓	✓	✓	✓	✓	✓	x	x	x	x	x
Contingency	✓	✓	✓	✓	✓	✓	✓	x	x	x	x	x
Enhancement	x	x	x	x	x	x	x	x	✓	✓	✓	✓

Fig. 1 : Purposes of MRD of Different Audience

Task-Orientation

Task orientation, which is attributed to system development methodology in software engineering, is broadly defined as an approach to MRD by which the accomplishment that describes the process and results of each task performed during each phase development of the simulation project are recorded and stored in documentation repository while the project is being performed sequentially or iteratively. This is more specifically referred as 'task documentation'. The individual groups of records are referred to as 'task documents' and the information embedded in task documents are referred to as 'task details'. 'Phase-end documentation' is described as the vehicle of communication that accumulates and arranges task documents to fulfil the needs of particular audience or individual concerned about the project, by retrieving appropriate task documents from the documentation repository.

Reusable Model Elements

Reusable model elements are primarily physical, information and control elements that are separated by the concept proposed by Bhuskute et al. (1992), Pratt et al. (1994), and Delen et al. (1996) to perform one-to-one function. But, for making the documentation an integral activity, the concept has been extended further to logical and data elements to capture model data and logics, which cannot be ascertained with the use of information and control elements alone and to separate model data from model logic.

- **Physical Elements:** A physical element is an object with a tangible correspondent in a real world. If the modellers' primary focus on the object is its physical characteristics, then the object is considered as physical element. Generic constituents of physical elements are processors, storages, transporters and conveyors. Separating physical elements from others allows the user to begin the documentation independently with easily identifiable information such as system layout.
- **Information Elements:** If the modellers' primary focus on the object is its information characteristics, then the object is designated as an information element. They are typically, arrive, depart, process, transport, convey elements supported with auxiliary elements like sets, batching, splitting, etc. As the entity flow through the system can be captured with the use of primary information elements, it allows the user to build up the model without intervention of physical/control element, but later to couple with them.
- **Control Elements:** Control elements signal to take appropriate action or control the model behaviour associated with physical and information elements. Decision, distribution in/out, ranking, priority and selection are considered as formalized control elements. Separation of these elements from others allows the user to describe explicitly how the model behaviour is controlled to achieve the expected results.

- **Logical Elements:** Logical element describes the logics of the executable model that can not be described with physical, information and control elements. Separation of these elements offers great capability in model exchange process with different simulation software.
- **Data Elements:** Data elements like process times, resource cycle data, other assign attributes, variables, expressions, etc. allow the user to record model data for documentation purpose. But, the data recorded with specified format allows transferring them into executable model. Such separation facilitates the modeller to build the model (or document the model) without interference of model data.

Task Documents

Fig. 2 shows the revised simulation project life cycle, containing major phases, sub-phases, primary tasks and task documents (from left to right).

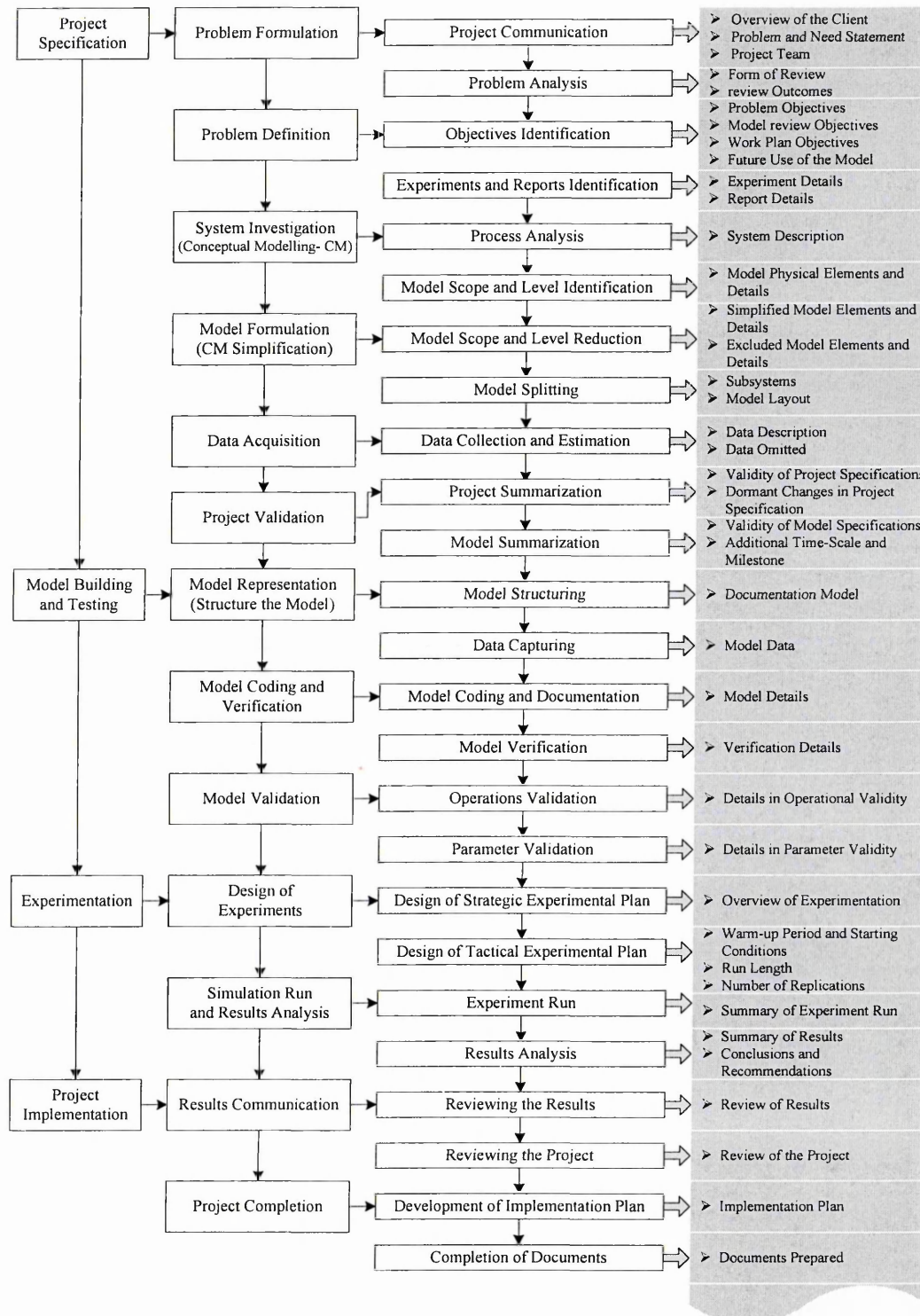


Fig. 2 : Task Documents in MRD

Documentation Model

Documentation model (DM) is the graphical representation of the executable model and is constructed with predefined generic reusable model elements. The technique adopted to represent DM is neutral technique – flowcharts – for making a provision to translate it into software dependent executable model. The graphical symbols corresponds to various types of reusable elements in DM are mainly the customised basic symbols in flowcharts and are composed into element library (a stencil in Visio). These elements are embedded with its own user properties – attributes and/or variables – that are collected at the time of using the element as user entries or are defined as identities, the values of which are placed interacting with other elements.

An example given in Fig. 3 shows the ‘process’ element (master - information type), what customer properties expected from the user, how customer properties are arranged to display the element information and what options that user has in selecting the appropriate function (process/delay) to suit the point of application. These properties of the elements are primarily utilized to present the description required for documentation, maintaining uniformity and consistency of the description throughout any project. But the same properties are employed to produce a ‘Common Database’, enabling to translate the documentation model into implementation model.

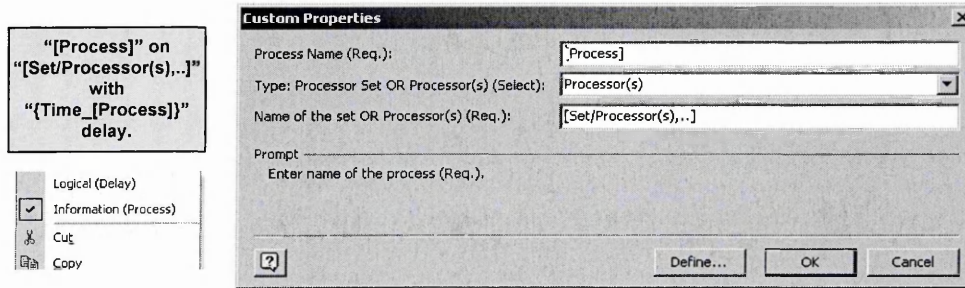


Fig. 3 : Process Element (Master)

Tree-View, Immediate Display Screen, Check-Boxes and Highlights of Task Documents

Figs. 4 and 5 show the features incorporated to promote the task documentation in MRD system for a ‘new user’ and a ‘regular user’ respectively. The end branches of the tree - task documents – act as a menu item to display the pre-structured forms that are required to be updated with task details; and to make possible other interactive features for making communication among task documents in interfaced software; depending on the nature of details.

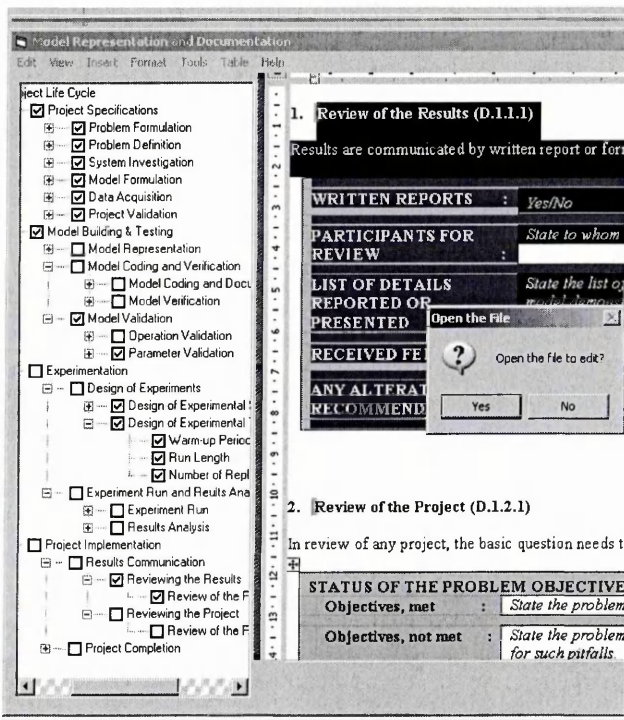


Fig. 4 : Features for Handling Task Documents
(For New User)

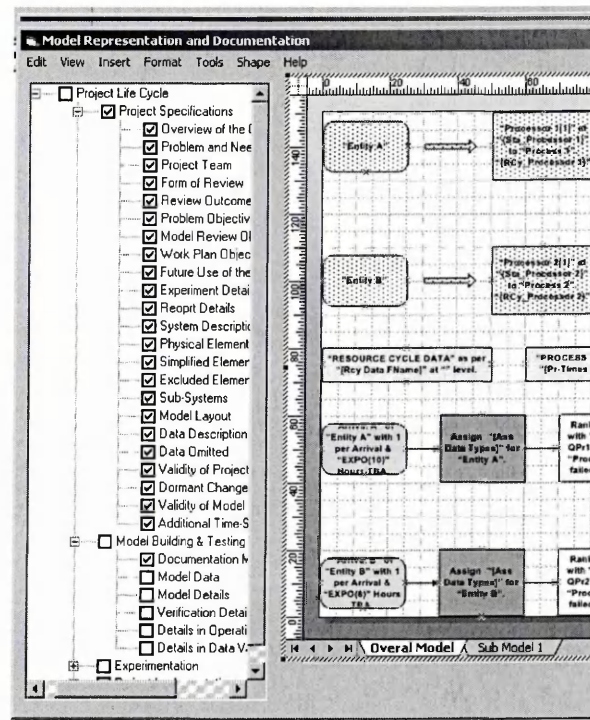


Fig. 5 : Features for Handling Task Documents
(For Regular User)

Interfaces for Data Sharing

Fig. 6 shows one of the interfaces developed between Word and Visio for data sharing in MRD system. For example, the model layout details, recorded in Word text file, are presented to the documentation model in Visio, through the interface developed with VB. The documentation model, represented in Visio, is then transferred into Word text file for recording purpose. Although the data in two locations are not in the same format, such hierarchical communication formulates the MRD system more effectively.

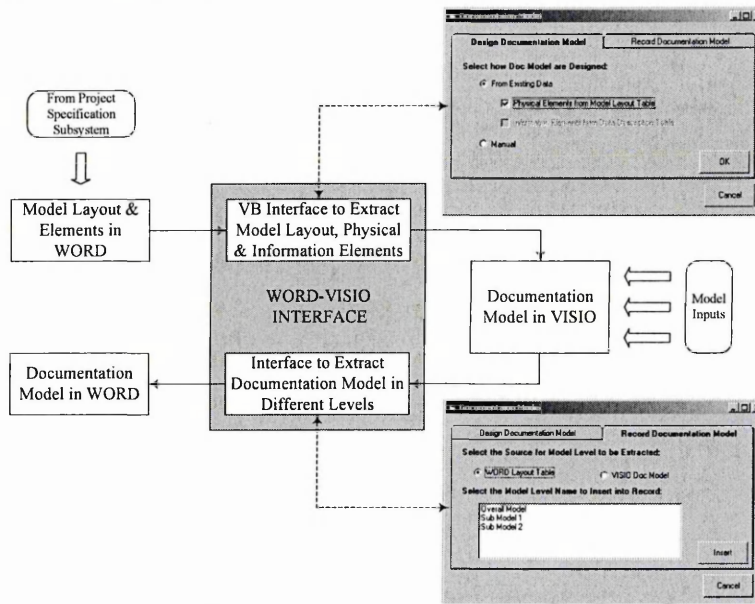


Fig. 6 : VISIO-WORD Interface for Hierarchical Communication

Interfaces for Document Preparation

Fig. 7 shows the screen of the interface provided to the user to handle the documents preparation task. The user is given an opportunity to select task documents for both purposes and audiences in preparation of documents. With the choice of particular type of purpose and audience, the interface displays suggested task documents appropriately as checked items in the tree view. However, due to variation of significance of task documents from project to project, the user should be given a flexibility to alter (add or remove task documents) final document appropriately or to prepare documents for his/her own choice.

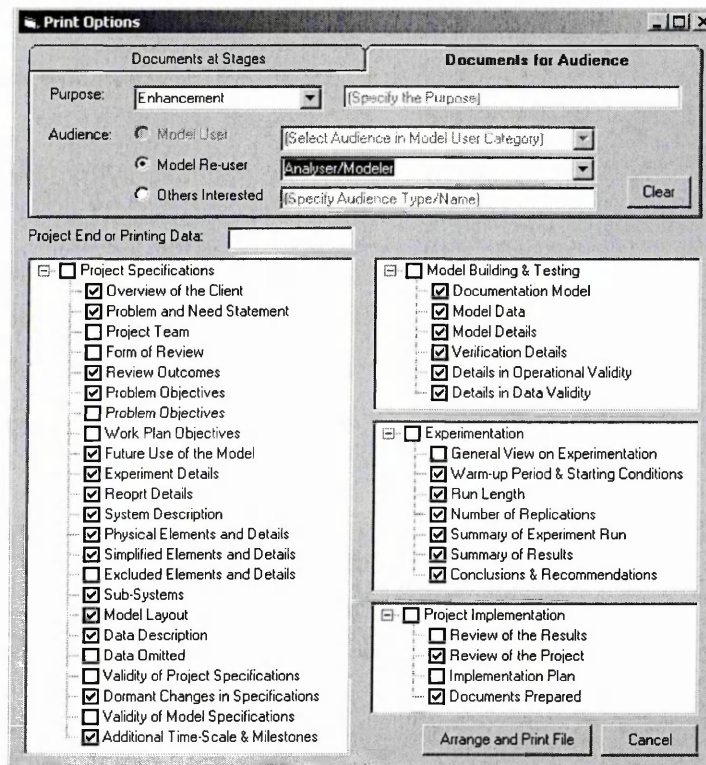


Fig. 7 : User Interface for Document Preparation

Common Database

Even with neutral model elements and technique, model translation cannot be performed without capturing details of individual elements and making a database of documentation model as medium of model transfer. Therefore, the common database is a collection of details of each element in the documentation model. It contains location details of model elements, user properties embedded in model elements and the data produced interacting with other elements. Those details are stored with standardized internal structure so that the documentation model could be translated into executable model of the simulation that is intended to use.

Usefulness of the common database is not limited to model exchange. It also provides necessary details for other task documents such as model data, verification elements. In respect of recording model data, the data element offers major contribution interacting with the documentation model database. Although, verification is associated with the software dependent executable model, producing the element list from a documentation model or its database presents a common layout for any project as the documentation model emphasizes each element that performs one-to-one function, thus avoiding the possibility of inattention or unawareness of individual elements.

Standard MRD System

Effective development and implementation of a documentation system is contingent, to a great extent, on the establishment of standard documentation structure that specify common layout, systematically linked details of items resulting from each task and the use of standard notations (e.g. rules, statistical values). With such complete documentation standards, MRD system will become standard MRD system that will offer more benefits to the simulation community as it attempts to establish a generic framework for the modeller to employ to manage and control the simulation project, while keeping the documentation in focus. Further, it improves the model exchangeability, interoperability and reusability with integrated MRD system. Fig. 8 illustrates how MRD process is integrated with simulation project through such standards and the potential benefits from such a standard system.

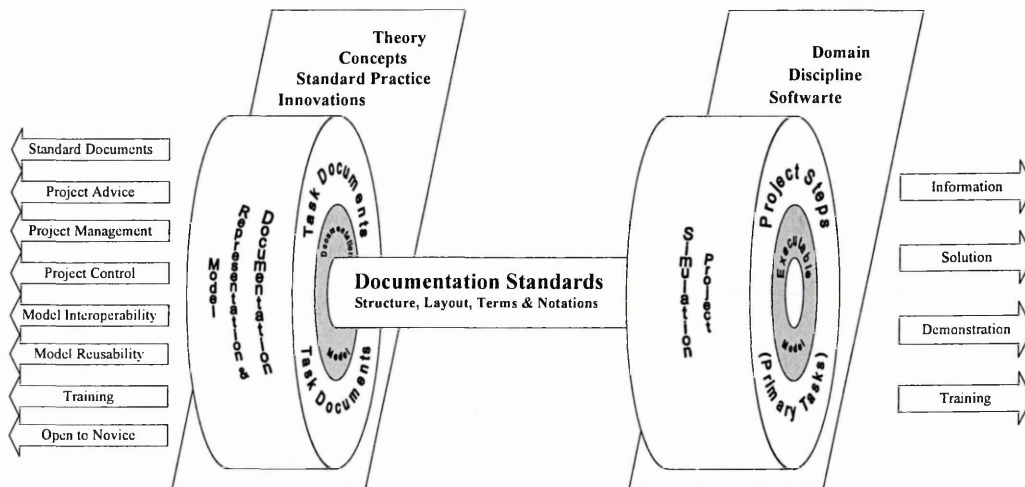


Fig. 8 : Standard MRD Process and Simulation Project

As can be seen in the above figure, when a simulation project begins to roll over the different domain and different simulation software, the MRD begins to turn over the simulation fundamentals for documentation purpose. If the two processes are coupled or integrated through documentation standards, then the two processes could be considered as a single process, offering more benefits to the simulation community.

ANNEXURE “C” – CASE STUDY: AN EXAMPLE FOR SIMULATION PROJECT DOCUMENT

MODEL REPRESENTATION & DOCUMENTATION ILLUSTRATION

PHASE-END DOCUMENT

From 1st February 2004 To 30th April 2004

1. Overview of the Client

The form below presents where the project is implemented, what future exception that the client has, and which level of expertise available for client.

CLIENT DETAILS:	
Location :	MRD Company, 77, Vincent Road, Sheffield, S7 1BW
Contact:	MRD@Doc.Com
BUSINESS GOALS :	The Company has an intention to improve the manufacturing and assembly plant in more flexible environment to expand their business with customers in other local and international market.
SYSTEM EXPERTISE :	System expertises are available for investigation of existing system.
SIMULATION EXPERTISE :	Company has knowledgeable members in simulation concepts for participating the project and future expansion.

Form 1 : Overview of the Client

2. Problem and Need Statement

The form shows the problem that the client solution seeks for, proposed solution technique(s), potential constraints for the project and expected benefits.

PROBLEM DESCRIPTION :	MRD Company assembles car engines for orders. The assembly requires seven different types of parts that are supplied by different suppliers. Three parts, Part A, Part B and Part C, that are required, one from each for assembly, need alterations before assembly operation and thus the parts arrived to manufacturing division are despatched to assembly division after their alterations. Other parts arrives assembly department directly for assembly operations. The resources and operations involve with each part in manufacturing division and the assembly operations are illustrated in Annex 1. The company needs to achieve 100 assembled products per week. But, it has never achieved the target; in fact, throughput is only just over 60% from the expected target. The Company is currently functioning 8 hour and 5 days normal working hours. The management believes that less throughput problem is a result of overload of capacity due to insufficient resources.
SOLUTION TECHNIQUE(S) :	The management believes that schedule of additional shifts to selected operations in both manufacturing and assembly divisions And/Or increase of resources capacity would increase the throughput.
ANY CONSTRAINTS :	If any shift is suggested, it should apply for complete subsystem like drilling centre, machining centre. The solution should be seek and implemented within 3 months to sustain the existing customers.
EXPECTED BENEFITS :	Increase in throughput of products. Opportunity to study the performance each division for future expansion.
References/ Appendix :	Annex 1: a. Manufacturing Operations; b. Assembly Operations (But not included in this example).

Form 2 : Problem and Need Statement

3. Project Team

Team members of the project team with their descriptions, the role expected from individual and the reasons for selections are presented in the form below.

TEAM MEMBER	DESCRIPTION	ROLE	REASON (S) FOR SELECTION
The Client/Nominator(s)	<i>Mr. Client, Director in manufacturing.</i>	<i>Direction, guiding and advising.</i>	<i>Authority for decision making.</i>
Team/Project Leader	<i>Mr. Team Leader, External consultant</i>	<i>Coordination of the project and modelling support.</i>	<i>Simulation expert.</i>
Model Author 1	<i>Mr. A. Model Author, Simulation expert</i>	<i>System investigation, model building, verification, validation.</i>	<i>Simulation expert and programmer.</i>
Model Author 2	<i>Mr. B. Model Author, System planning engineer</i>	<i>System investigation, model building, verification, validation, implementation.</i>	<i>In-house member with sound system and simulation knowledge.</i>
System Supporter & Data Collector (1)	<i>Mr. A. SysSup DataColl, Supervisor</i>	<i>System supporter for machine cell and data collection of part B and in machine cell.</i>	<i>Knowledgeable on operations of machine cell.</i>
System Supporter & Data Collector (2)	<i>Mr. B. SysSup DataColl, Supervisor</i>	<i>System supporter for rework centre and data collection of part C and in rework centre.</i>	<i>Knowledgeable on operations of rework centre.</i>
System Supporter & Data Collector (3)	<i>Mr. C. SysSup DataColl, Forman</i>	<i>System supporter for drilling centre and data collection in drilling centre.</i>	<i>Knowledgeable on operations of drilling centre.</i>
System Supporter & Data Collector (4)	<i>Mr. D. SysSup DataColl, Forman</i>	<i>System supporter for departing & inspection centre and Data collection of part A and departing region.</i>	<i>Knowledgeable on operations in departing and inspection region.</i>
Modelling Supporter(s)	<i>Not available.</i>	<i>Not available.</i>	<i>Not available.</i>
Model Executor	<i>Mr. Model Executor, Factory engineer</i>	<i>Experimentation, reporting, and implementation support.</i>	<i>Knowledgeable person on system and keen on simulation systems.</i>

Form 3 : Team Members

4. Form of Review

The form(s) below presents what and how specific issue(s) of the problem was reviewed and the outcomes from the review.

METHOD OF REVIEW :	<i>Workshop & Discussion</i>
PERSONALS PARTICIPATED:	<i>The client, team leader, modellers, managers in production planning division, engineers in assembly division, managers in supply and receiving department.</i>
ISSUES REVIEWED :	<i>Influence of assembly operation on the throughput. (To clarify any effect of exiting layout and operations on the output of assembly product that seems to have no effect on the problem in throughput).</i>
REVIEW SUMMARY :	<i>Inefficient use of assembly resources portrays that the operations of assembly division has no effect on the exiting throughput and it has the capability to increase the output by 30-40% with no cost.</i>

Review (1)

METHOD OF REVIEW :	<i>Workshop & Discussion</i>
PERSONALS PARTICIPATED:	<i>Team leader, modellers, system supporters, engineers in assembly centre, milling, reworking, departing and inspection centres; managers in production planning division.</i>
ISSUES REVIEWED :	<i>Influence of material handling system on the problem domain. (To clarify the issues rose regarding the material handling during the brainstorming sessions and interviews held with relevant parties).</i>
REVIEW SUMMARY :	<i>Existing material handling transportation system has a little influence on the current issues in individual centres or product output from the manufacturing system, if efficiency of the transportation system remains in the existing level.</i>

Review (2)

Form 4 : Form of Review

5. Review Outcomes

The form below delivers any alteration suggested to the problem with alternative solutions; any constraints & assumptions made and expected real benefits.

ALTERATION TO PROBLEM :	<i>None.</i>
ALTERNATIVE SOLUTION :	<i>Expansion of resources may be applicable to CNC grinder, boring machine, milling machines only. Introduction of shift may be applied to drilling and/or departing centres only.</i>
CONSTRAINTS :	<i>Use of spare grinder, which has lower performance than the CNC grinder.</i>
ASSUMPTIONS :	<i>No influence of assembly operation on the problem domain. Effect of material handling system in manufacturing is negligible.</i>
BENEFITS ACHIEVABLE :	<i>Any improvement to output or throughput of the manufacturing division. The study of performance of manufacturing division.</i>

Form 5 : Review Outcomes

6. Problem Objectives

The form below present all ranked objectives related the problem, including what to be achieved, measurement of achievement and under what constraints achievements are made.

ACHIEVEMENT	MEASUREMENT	CONSTRAINT	RANK
<i>Increase of product output or throughput</i>	<i>At least 15% (from existing level)</i>	<i>Use of additional resource (Spare grinder) with CNC grinder.</i>	<i>1</i>
<i>Increase of product output or throughput</i>	<i>5-10% (from existing level)</i>	<i>Increase of one shift to selected centres (drilling or departing) or further increase in another resource (miller, boring machine or inspector for inspection).</i>	<i>2</i>
<i>Study the performance of individual operations in manufacturing division.</i>	<i>None</i>	<i>None</i>	<i>3</i>

Form 6 : Problem Objectives

7. Model Review Objectives

The record below show the rough estimates initially set on model run speed and type of visual display expected in model run.

RUN SPEED OF THE MODEL :	<i>Run speed should be moderate to display the individual performance, but taking into account the time constraint imposed at the beginning of the project.</i>
VISUAL DISPLAYS :	<i>Due to time restriction, a minimum number of graphical displays are to be included, but taking the management desire of understanding the system performance into account.</i>

Form 7 : Model Review Objectives

8. Work Plan Objectives

The table below presents the work schedule, and review stages of the project including their timescale with additional ones and persons responsible for.

MAJOR ACTIVITY (WORK/REVIEW)	PROPOSED TIME-SCALE	ADDITIONAL MILESTONE	PERSANAL INVOLVEMENT
<i>Project and Model Specifications</i>	<i>28th February 2004</i>	<i>5th March 2004</i>	<i>Team leader, modellers and system supporters</i>
<i>Documentation and Executable Mode</i>	<i>21st March 2004</i>	<i>28th March 2004</i>	<i>Modellers</i>
<i>Data Collection</i>	<i>21st March 2004</i>	<i>28th March 2004</i>	<i>Data collectors</i>
<i>Model Validation</i>	<i>31st March 2004</i>	<i>5th April 2004</i>	<i>Modellers, model executor</i>
<i>Experimentation</i>	<i>12th April 2004</i>	<i>15th April 2004</i>	<i>Model executor</i>
<i>Implementation</i>	<i>30th April 2004</i>	<i>5th May 2004</i>	<i>Implementation team including modeller 2</i>

Form 8 : Work Plan Objectives

9. Future Use of the Model

Usage of the model for future purposes is given in the following table.

SHORT-TERM USE :	<i>None.</i>
LONG-TERM USE :	<i>Expect to use the model for expansion of the manufacturing division and more specifically development of individual manufacturing centres (machining cell, rework cell, drill centre, inspection centre).</i>

Form 9 : Future Use of the Model

10. Experiment Details

The table below gives what variables are changed, what range of values they have, how values are fed, what assumption made, if available, for experimentation; and what problem objectives are met from each experiment.

OBJECTIVE	EXPERIMENTAL FACTORS	FACTOR LEVELS	METHOD OF DATA ENTRY	ANY ASSUMPTION
Increase throughput	Mill Capacity	1 - 2	Menu driven	Capacity of operator attached to miler also change accordingly.
	Borer Capacity	1 - 2	Menu driven	Capacity of operator attached to borer also change accordingly.
	CNC Grinder Capacity	1 - 2	Menu driven	Capacity of operator attached to grinder also change accordingly.
	Spare Grinder	0 - 1	Model Code	Spare grinder is allocated, only if necessary.
	Spare Inspector	0 - 1	Model Code	Spare inspection is allocated, only if necessary.
	Drilling centre shift	Fully Flexible	Menu driven	None.
	Departing centre shift	Fully Flexible	Menu driven	None.

Form 10 : Experiment Details

11. Report Details

The table shows what and how values from each experiment are to be reported and how they are viewed in presentation of results.

EXPERIMENT	VALUES TO BE REPORTED	METHOD OF REPORTING	METHOD OF VIEWING
Mill Capacity Borer Capacity CNC Grinder Capacity Spare Grinder Spare Inspector Drilling centre shift Departing centre shift	Weekly model Throughput	Individual average	Interactive Data file Dynamic displays
	Weekly Sub-model throughput	Cumulative total	
	Resource utilization	Percentages	
	Waiting time	Time series	
	Number in/out	Average values	
	Number waiting		
	Cycle time		

Form 11 : Report Details

12. System Description

The major functional areas included into the system of the study and any other areas omitted from the system with the reasons for omission are given below.

OVERALL SYSTEM :	Manufacturing area and assembly area.	
MAJOR AREAS SELECTED :	Manufacturing area that includes machining, rework, drilling and departing centres (Annex 1: a. Manufacturing area).	REASON FOR SELECTION : Manufacturing area is the major contributor for low throughput.
SPECIAL AREAS OMITTED :	Assembly area (Annex 1: b. Assembly area).	REASON FOR OMISSION : It is assumed that the performance of has no or a little effect on low throughput.
References/Appendix :	Annex 1: a. Manufacturing Area; b. Assembly area. (But not included in this example).	

Form 12 : System Description

13. Physical Elements and Details to be modelled

The following table presents physical elements to be included into the model and checklist of their level of details together with reasons or any assumption made in respect of each element.

TYPE OF PHYSICAL ELEMENT	NAME OF THE PHYSICAL ELEMENT	CONTRIBUTION LEVEL (MAJOR/MINOR)	LEVEL OF DETAILS					ASSUMPTIONS OR ANY SPECIAL REASON FOR MODELLING THE ELEMENT/DETAILS
			INDIVIDUAL OPERATION?	INDIVIDUAL SET-UPS?	SCHEDULE/SHIFTS?	FAILURES/BREAKDOWN?	
ENTITY	Part A	Major						None.
ENTITY	Part B	Major						
ENTITY	Part C	Major						
PROCESSOR	Saw	Minor	No	No	No	No	No	Processing on Part A at the beginning (before CNC grinding) has output to the expected level.
PROCESSOR	Puncher	Minor	No	No	No	No	No	
PROCESSOR	Lathe	Minor	No	No	No	No	No	

PROCESSOR	Threader	Minor	No	No	No	No	No	No	No		
PROCESSOR	Cell Loader	Major	N/A	N/A	Yes	Yes	Yes	No	No		Cell Loader loads Part B to different processors.
PROCESSOR	CNC	Major	Yes	Yes	Yes	Yes	Yes	Yes	No		Individual set-up for Part B loading for CNC and grinder with cell loader is required.
PROCESSOR	Grinder	Major	Yes	Yes	Yes	Yes	Yes	Yes	No		Polishing on Part B is performed with cell loader.
PROCESSOR	Polisher	Major	Yes	Yes	No	Yes	Yes	Yes	No		None.
PROCESSOR	Face Miller	Major	Yes	Yes	No	No	Yes	Yes	No		
PROCESSOR	Surface Tester	Major	Yes	Yes	No	No	Yes	Yes	No		
PROCESSOR	Surface Hardener	Major	Yes	Yes	No	No	Yes	Yes	No		
PROCESSOR	Miller	Major	Yes	Yes	No	No	Yes	Yes	Yes		Shift operations or expansion of resources are essential due to high time and volume on Parts B & C for experimentation purpose.
PROCESSOR	DrillB1	Major	Yes	Yes	No	No	Yes	Yes	No		
PROCESSOR	DrillB2	Major	Yes	Yes	No	No	Yes	Yes	No		
PROCESSOR	DrillBC	Major	Yes	Yes	No	No	Yes	Yes	No		
PROCESSOR	DrillC1	Major	Yes	Yes	No	No	Yes	Yes	No		
PROCESSOR	DrillC2	Major	Yes	Yes	No	No	Yes	Yes	No		
PROCESSOR	Borer	Major	Yes	Yes	No	No	Yes	Yes	No		
PROCESSOR	CNC Grinder	Major	Yes	Yes	No	No	Yes	Yes	Yes		Shift operations or expansion of resources are essential due to high volume on Parts A, B & C and as experimentation purpose. Failure is considered due to regular failing of resources and high processing time.
PROCESSOR	Spare Grinder	Major	Yes	Yes	No	No	Yes	Yes	Yes		
PROCESSOR	Deburer	Major	Yes	Yes	Yes	Yes	Yes	Yes	Yes		
PROCESSOR	Loader Inspector	Major	Yes	Yes	No	No	Yes	Yes	Yes		
PROCESSOR	Spare Inspector	Major	Yes	Yes	No	No	Yes	Yes	Yes		

Form 13 : Physical Elements and Details to be Modelled

14. Simplified Elements and Details

The table below shows what elements/details were simplified, what elements/details were replaced with simplified elements/details and why, together with any assumptions made in simplification.

ELE & DETAILS TO BE GROUPED/ SUSTITUTED	REPRESENTING ELE & DETAILS	ASSUMPTIONS AND REASON S	ANY RESULTS BE AFFECTED
Saw, puncher, lathe, threader and their processes.	"Processor Group" with "Process delay".	No much influence on solutions techniques or objectives.	Individual queues, performance of processors.

Form 14 : Simplified Element and Details

15. Excluded Elements and Details

Elements and details to be excluded from the model and reason for such exclusion with potential output losses are presented in the following table.

ELE & DETAILS TO BE EXCLUDED	REASON S AND ASSUMPTIONS	ANY RESULTS BE AFFECTED
<i>Individual operators for setting up of each major processor.</i>	<i>Individual performance of operators is not essential and they are assumed freely available to provide service.</i>	<i>Performance of individual operators. Throughput time of each part.</i>
<i>Transportation details between sub-systems or sub-models.</i>	<i>Transportation has minor influence on problem objectives and is assumed freely available with no time.</i>	<i>Throughput time of each part.</i>

Form 15 : Excluded Element and Details

16. Subsystems

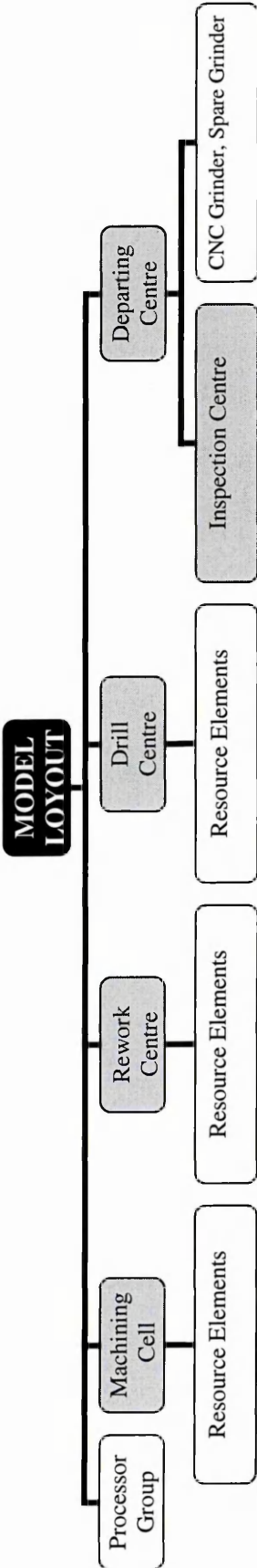
The details of subsystems, how they are linked, and any assumptions associated in isolation process are presented in the table below.

NAME OF SUBSYSTEM	ELEMENTS & NEXT LEVEL SUBSYSTEMS	INPUTS TO SUBSYSTEM	OUTPUTS FROM SUBSYSTEM	TECHNIQUES AND ASSUMPTIONS
<i>Machine Cell</i>	<i>CNC, Cell Loader, Grinder, Polisher</i>	<i>Part B - In</i>	<i>Part B - Out</i>	<i>Flow approach. Ignore transportation.</i>
<i>Rework Centre</i>	<i>Face Miller, Surface Tester, Surface Hardener</i>	<i>Part C - In</i>	<i>Part C - Out</i>	<i>Flow approach. Ignore transportation.</i>
<i>Drill Centre</i>	<i>Miller, DrillB1, DrillB2, DrillC1, DrillC2, DrillBC, Borer</i>	<i>Part B & C - In</i>	<i>Part B & C - Out</i>	<i>Flow approach. Ignore transportation.</i>
<i>Departing Centre</i>	<i>Inspection Centre CNC Grinder, Spare Grinder</i>	<i>Part A, B & C - In</i>	<i>Part B & C - Out</i>	<i>Flow approach. Ignore transportation.</i>
<i>Inspection Centre</i>	<i>Deburer, Loader Inspector, Spare Inspector</i>	<i>Part B & C - In</i>	<i>Passed Parts B&C</i>	<i>Flow approach.</i>

Form 16 : Excluded Element and Details

17. Model Layout

The following figure and table illustrates the relationships of subsystems (sub-models) and physical elements with the overall model, followed by individual physical elements of each level.



Model Layout

MODEL LEVEL NAME	MODEL LEVEL NUMBER	NAMES OF SUB-MODELS	NAMES OF PROCESSORS	NAMES OF STORAGES	NAMES OF TRANSPORTERS	NAMES OF CONVEYORS
OverallModel	1	MachineCell, ReworkCentre, DrillCentre, DepartingCentre	ProcessorGroup			
Machining	1.1		CNC, CellLoader, Grinder, Polisher			
Reworking	1.2		FaceMiller, Surface Tester, Surface Hardner			
Drilling	1.3		Miller, DrillB1, DrillB2, DrillC1, DrillC2, DrillBC, Borer			
Departing	1.4	InspectionCentre	CNCGrinder, SpareGrinder			
Inspection	1.4.1		Deburer, LoaderInspector, SpareInspector			

Form 17 : Lavout Details

18. Data Description

The form below shows the data required for the model, data sources or how estimated, the purpose of data, input format, any assumptions made & who is responsible for collection.

TYPE OF ELEMENT	ELEMENT NAME	TYPE OF DATA ITEM	PURPOSE*	INPUT FORMAT*	SOURCE/HOW ESTIMATED	ALTERNATIVE SOURCES/WAYS	APPROXIMATE % ACCURACY	ANY ASSUMPTIONS	PERSON RESPONSIBLE*	FUTURE CHANGE EXPECTS?
PROCESS	Process Delay	PROCESS TIME	2	1	Passed data		2		4	No
PROCESSOR	Cell Loader	SCHEDULE	2,3	1	Passed data		2		1	No
PROCESSOR	CNC	SCHEDULE	2,3	1	Passed Data		2		1	No
PROCESSOR	Grinder	SCHEDULE	2,3	1	Passed Data		2		1	No
PROCESS	CNC Loading	PROCESS TIME	2	1	Supplier data		2		1	No
PROCESS	CNC Machining	PROCESS TIME	2	1	Passed data		2		1	No
RANKING	Ra-CNC Loading	RA-CRITERIA	1	2	Passed data		2		1	No
PROCESS	Grinder Loading	PROCESS TIME	2	1	Passed data		2		1	No
PROCESS	Grinding	PROCESS TIME	2	1	Passed data		2		1	No
RANKING	RaGrinderLoading	RA-CRITERIA	1	2	Passed data		2		1	No
PROCESSOR	Polisher	SCHEDULE	2,3	1	Passed Data		2		1	No
PROCESS	Polishing	PROCESS TIME	2	1	Passed data		2		1	No
RANKING	Ra-Polishing	RA-CRITERIA	1	2	Passed data		2		1	No
PROCESSOR	Miller	SCHEDULE	2,3	1	Passed data		2		3	No
		FAILURE	2	1	Supplier data	Passed Data	2		3	No
PROCESS	Milling	PROCESS TIME	2	1	Passed data		2		3	No
PROCESSOR	DrillB1	SCHEDULE	2,3	1	Passed data		2		3	No
PROCESSOR	DrillB2	SCHEDULE	2,3	1	Passed data		2		3	No
PROCESSOR	DrillBC	SCHEDULE	2,3	1	Passed data		2		3	No
PROCESS	DrillingB	PROCESS TIME	2	1	Passed data		2		3	No
RANKING	Ra-DrillingB	RA-CRITERIA	1	2	Passed data		2		3	No
SELECTION	Se-DrillingB	SE-CRITERIA	1	2	Passed data		2		3	No
PROCESSOR	Borer	SCHEDULE	2,3	1	Passed data		2		3	No
		FAILURE	2	1	Supplier data	Passed Data	2		3	No
PROCESS	Boring	PROCESS TIME	2	1	Passed data		2		3	No
RANKING	Ra-Boring	RA-CRITERIA	1	2	Passed data		2		3	No

PROCESSOR		RA-PRIORITY	1	2	Passed data		2	3	No
	CNC Grinder	SCHEDULE	2,3	1	Passed data		2	4	No
		FAILURE	2	1	Supplier data	Passed Data	2	4	No
PROCESS	CNC Grinding	PROCESS TIME	2	1	Passed data		2	4	No
RANKING	Ra-CNC Grinding	RA-CRITERIA	1	2	Passed data		2	4	No
		RA-PRIORITY	1	2	Passed data		2	4	No
PROCESSOR	Spare Grinder	SCHEDULE	3	1	Proposed data		3	4	No
		FAILURE	3	1	Supplier data		2	4	No
PROCESS	Spare Grinding	PROCESS TIME	3	1	Proposed data		3	4	No
RANKING	Ra-SpareGrinding	RA-CRITERIA	3	2	Proposed data		3	4	No
PROCESSOR	Loader Inspector	SCHEDULE	2,3	1	Passed data		2	4	No
PROCESSOR	Deburer	SCHEDULE	2,3	1	Passed data		2	4	No
		FAILURE	2	1	Supplier data	Passed Data	2	4	No
PROCESS	Debur Loading	PROCESS TIME	2	1	Passed data		2	4	No
PROCESS	Deburring	PROCESS TIME	2	1	Passed data		2	4	No
RANKING	Ra-Debur Loading	RA-CRITERIA	1	2	Passed data		2	4	No
		RA-PRIORITY	1	2	Passed data		2	4	No
PROCESS	Inspection	PROCESS TIME	2	1	Passed data		2	4	No
RANKING	Ra-Inspection	RA-CRITERIA	1	2	Passed data		2	4	No
		RA-PRIORITY	1	2	Passed data		2	4	No
DISTRIBUTION	Di-Inspection	DI-VALUES	2	2	Passed data		2	4	No
RANKING	R-SpareInspection	RA-CRITERIA	3	2	Proposed data		3	4	No
		RA-PRIORITY	3	2	Proposed data		3	4	No
ARRIVAL	PartA Arrival	TB ARRIVALS	2	1	Passed data		2	4	No
SUB-MODEL	Drilling (B,C)	THROUGHPUT	4	n/a	Passed data		2	3	No
		THROU'PUT TIME	4	n/a	Passed data		2	3	No
SUB-MODEL	Departing(A,B,C)	THROUGHPUT	4	n/a	Passed data		2	4	No
		THROU'PUT TIME	4	n/a	Passed data		2	4	No
SYSTEM	Manufacturing	THROU'PUT TIME	4	n/a	Passed data		2	4	No
		THROUGHPUT	4	n/a	Passed data		2	4	No
KEYS *	PURPOSE		INPUT FORMAT		PER. RESPONSIBLE		ACCURACY		
	1.	For model representation.	1.	Menu driven.	1.	Data Provider1	1.	Very Good	
	2.	For model building.	2.	Model code.	2.	Data Provider2	2.	Good	
	3.	For experimentation.			3.	Data Provider3	3.	Fair	
	4.	For validation.			4.	Data Provider4	4.	Bad	

Form 18 : Data Description (Part of the list)

19. Data Omitted

The data items excluded from the model, the reasons for such exclusion, any assumptions made and potential results be affected display in the table below.

ELEMENT TYPE/NAME	DATA ITEMS OMITTED	REASON/ANY ASSUMPTION	ANY RESULTS BE AFFECTED
Cutting part A on saw machine.	Process time.	Non-availability of process time on cutting process makes it omit data.	As the saw machine make minor contribution to problem domain, the affect on omission can be ignored.

Form 19 : Data Omitted

20. Validity of Project Specifications

The form below present the project specifications validated, the participants in validation process and any alteration proposed to specifications.

PROJECT SPEC VALIDATED :	Problem objectives; model review objectives; work plan objectives; experiment details; report details; system descriptions; simplified elements and details; excluded elements and details; and data omitted
PARTICIPANTS FOR VALIDATION :	The client, team leader, modellers, system supporters, data providers, managers in production planning division, engineers in manufacturing and assembly division, managers in supply and receiving department.
PROPOSED ALTERATIONS TO PROJECT SPEC :	Reports on performance of different processors and subsystems in manufacturing are expected.

Form 20 : Validity of Project Specification

21. Dormant Changes in Project Specifications

The description below presents the proposed changes to specifications, how such changes are handled, and potential effects on model results due to suppression or delaying the suggested changes.

CHANGES PROPOSED TO SPECIFICATIONS	METHOD OF HANDLING CHANGES	ANY RESULTS BE AFFECTED
Reports expected for performance on the processors that process part A on saw, lathe, puncher and threder.	They are handled in future use of the model in expansion of the system.	Not much effect on the results as they offer minor contribution to the problem domain.

Form 21 : Dormant Changes in Project Specification

22. Validity of Model Specifications

The model specifications validated; the participants for model validation process and any alteration made to model specifications present in the form below.

MODEL SPEC VALIDATED :	Problem objectives; model review objectives; work plan objectives; experiment details; report details; system descriptions; physical elements & details to be modelled, simplified elements and details; excluded elements and details; subsystems; model layout; and data description.
PARTICIPANTS FOR VALIDATION :	The client, modellers, system supporters, data providers, managers in production planning division, engineers in manufacturing and assembly division.
PROPOSED ALTERATIONS TO MODEL SPEC :	None.

Form 22 : Validity of Model Specification

23. Additional Time-scale & Milestone

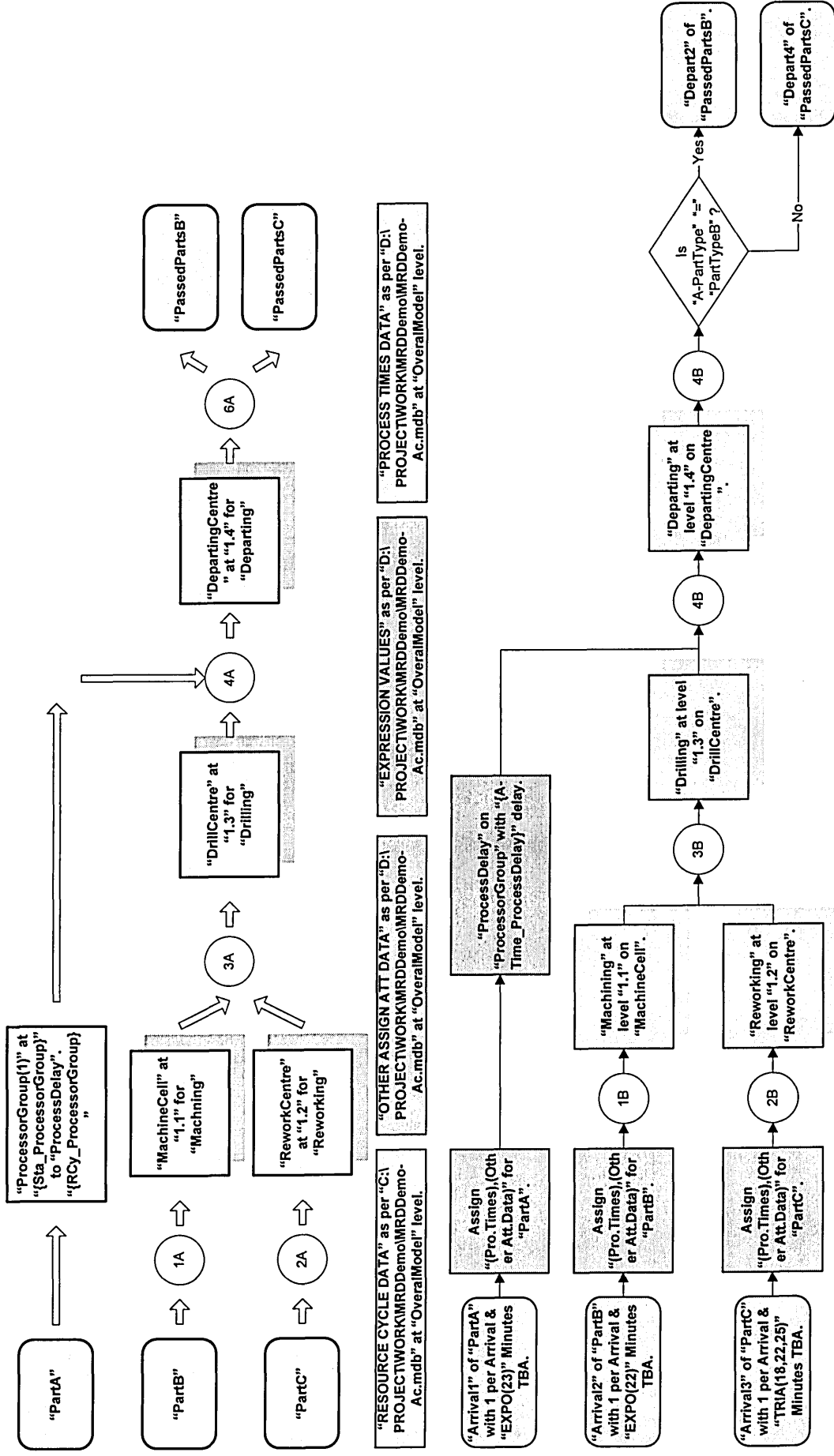
The table below presents time scales established and individual responsible for different sub models.

NAME OF SUB-MODEL	RESPONSIBLE PERSONAL	TIME SCALE
Machining & Reworking	Modeller 1 & 2	14 th March 2004
Drilling & Departing	Modeller 1 & 2	22 nd March 2004

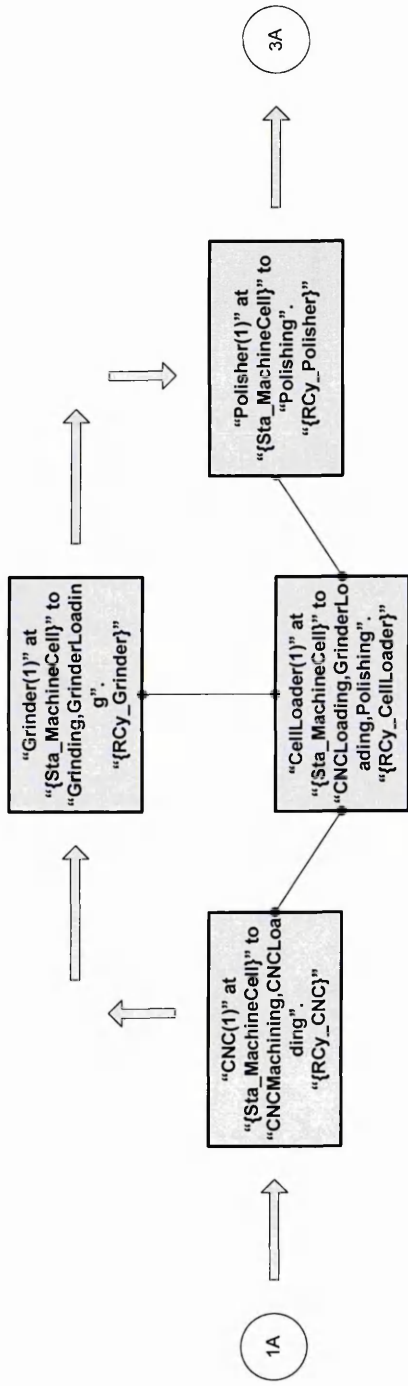
Form 23 : Dormant Changes in Project Specification

24. Documentation Model

The documentation model below presents the structure the model, its logic and makes the interfaces to model data.



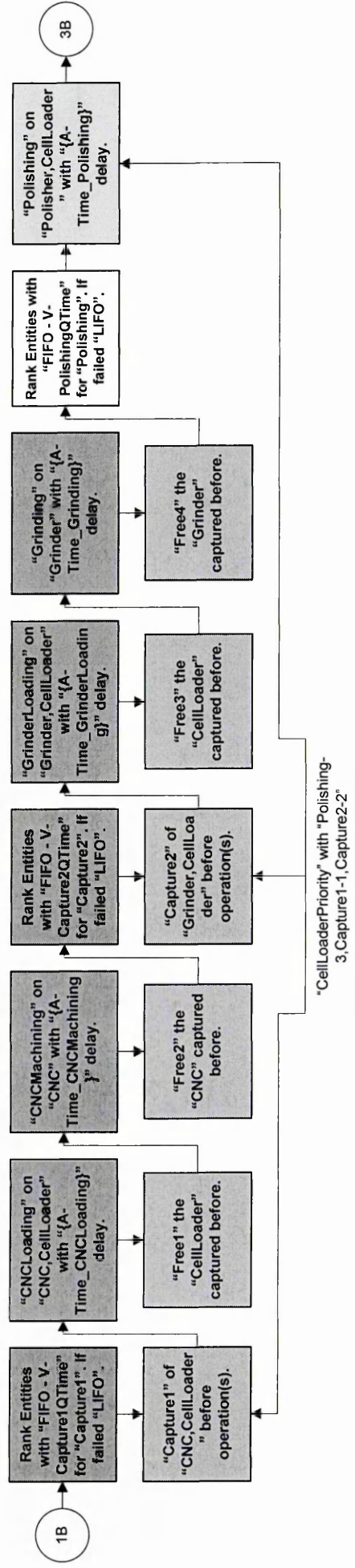
Overall Model



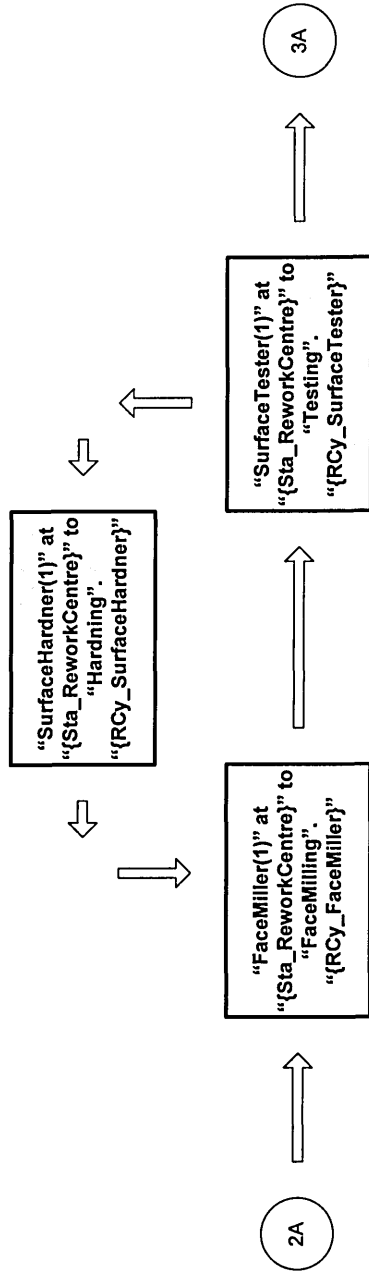
"RESOURCE CYCLE DATA" as per "C:\PROJECTWORK\MRDDemo\MRDDemo-Ac.mdb" at "Machining" level.

"PROCESS TIMES DATA" as per "C:\PROJECTWORK\MRDDemo\MRDDemo-Ac.mdb" at "Machining" level.

"OTHER ASSIGN ATT DATA" as per "C:\PROJECTWORK\MRDDemo\MRDDemo-Ac.mdb" at "Machining" level.



Sub-Model : Machining

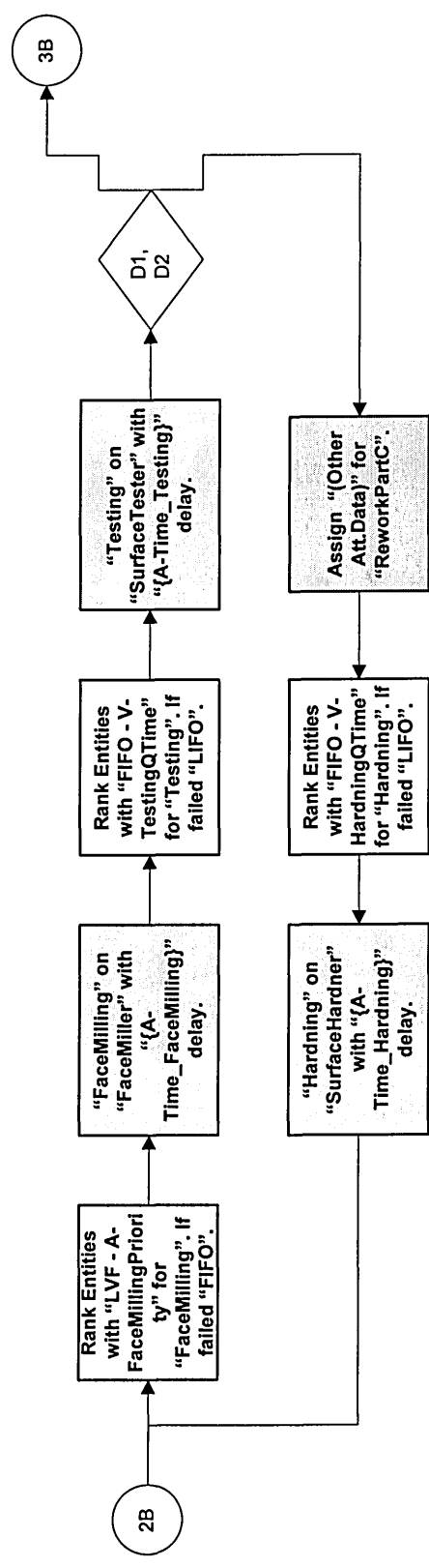


"RESOURCE CYCLE DATA" as per "C:\PROJECTWORK\MRDDemo\MRDDemo-Ac.mdb" at "Reworking" level.

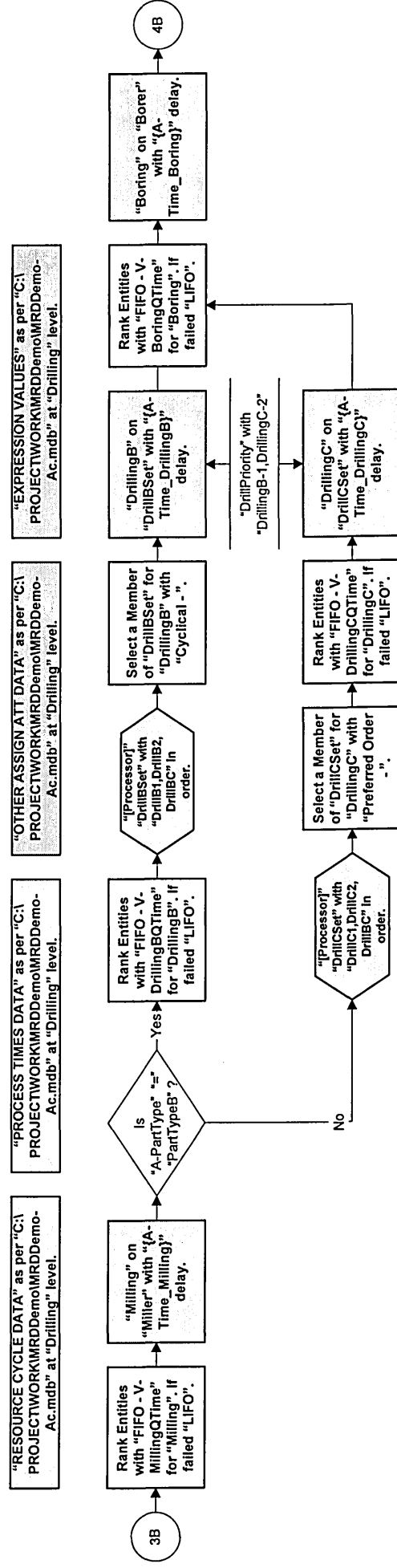
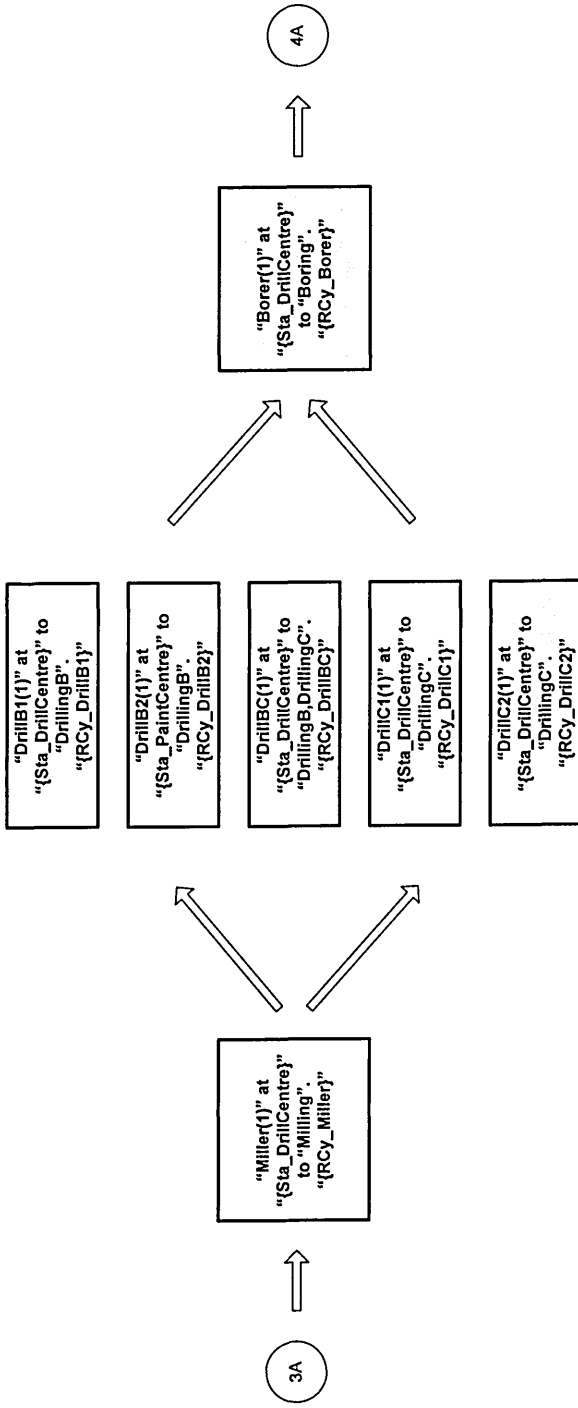
"PROCESS TIMES DATA" as per "C:\PROJECTWORK\MRDDemo\MRDDemo-Ac.mdb" at "Reworking" level.

"OTHER ASSIGN ATT DATA" as per "C:\PROJECTWORK\MRDDemo\MRDDemo-Ac.mdb" at "Reworking" level.

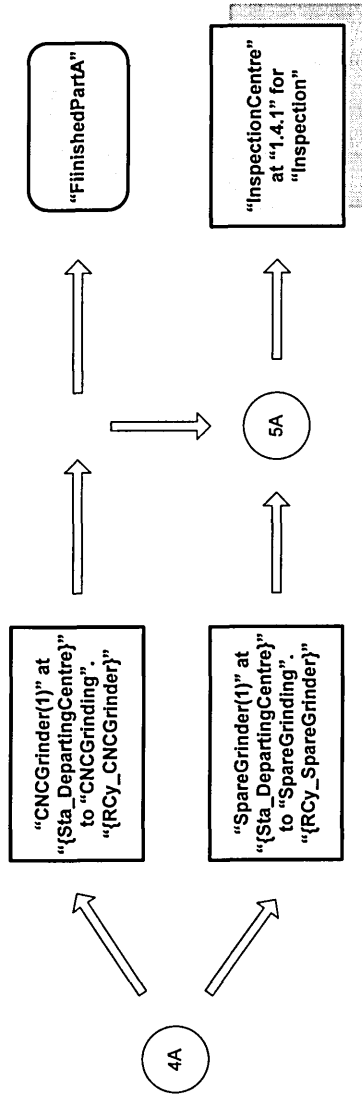
"EXPRESSION VALUES" as per "C:\PROJECTWORK\MRDDemo\MRDDemo-Ac.mdb" at "Reworking" level.



Sub-Model : Reworking



Sub-Model : Drilling

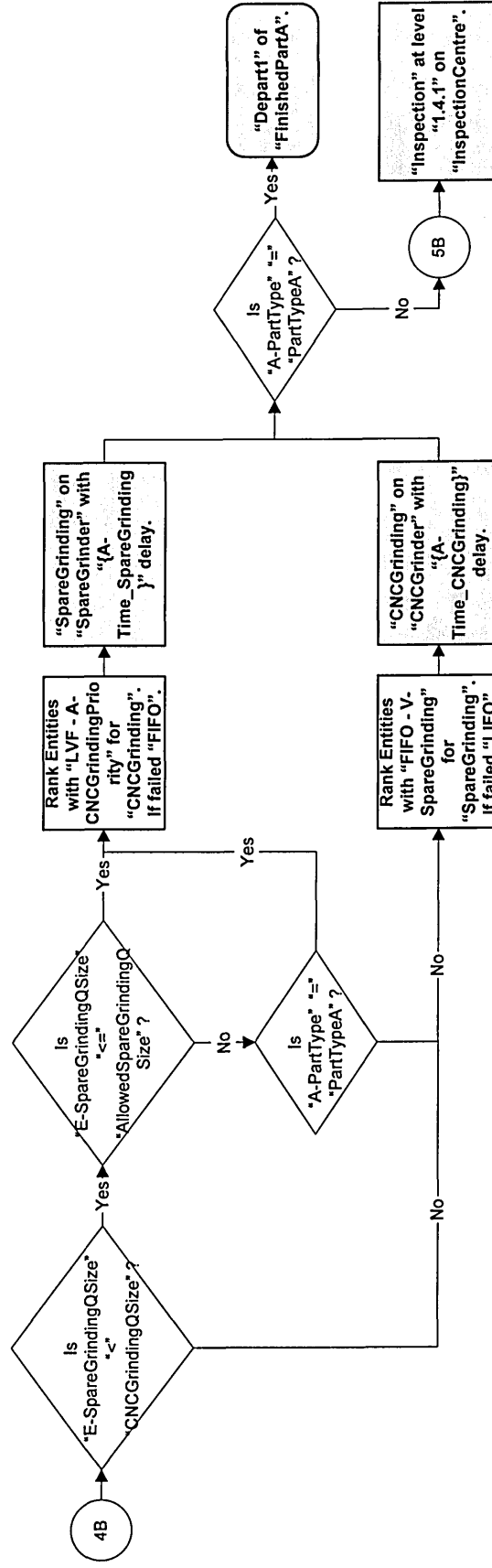


"RESOURCE CYCLE DATA" as per "C:\PROJECT\WORK\MRDDemo\MRDDemo-Ac.mdb" at "Departing" level.

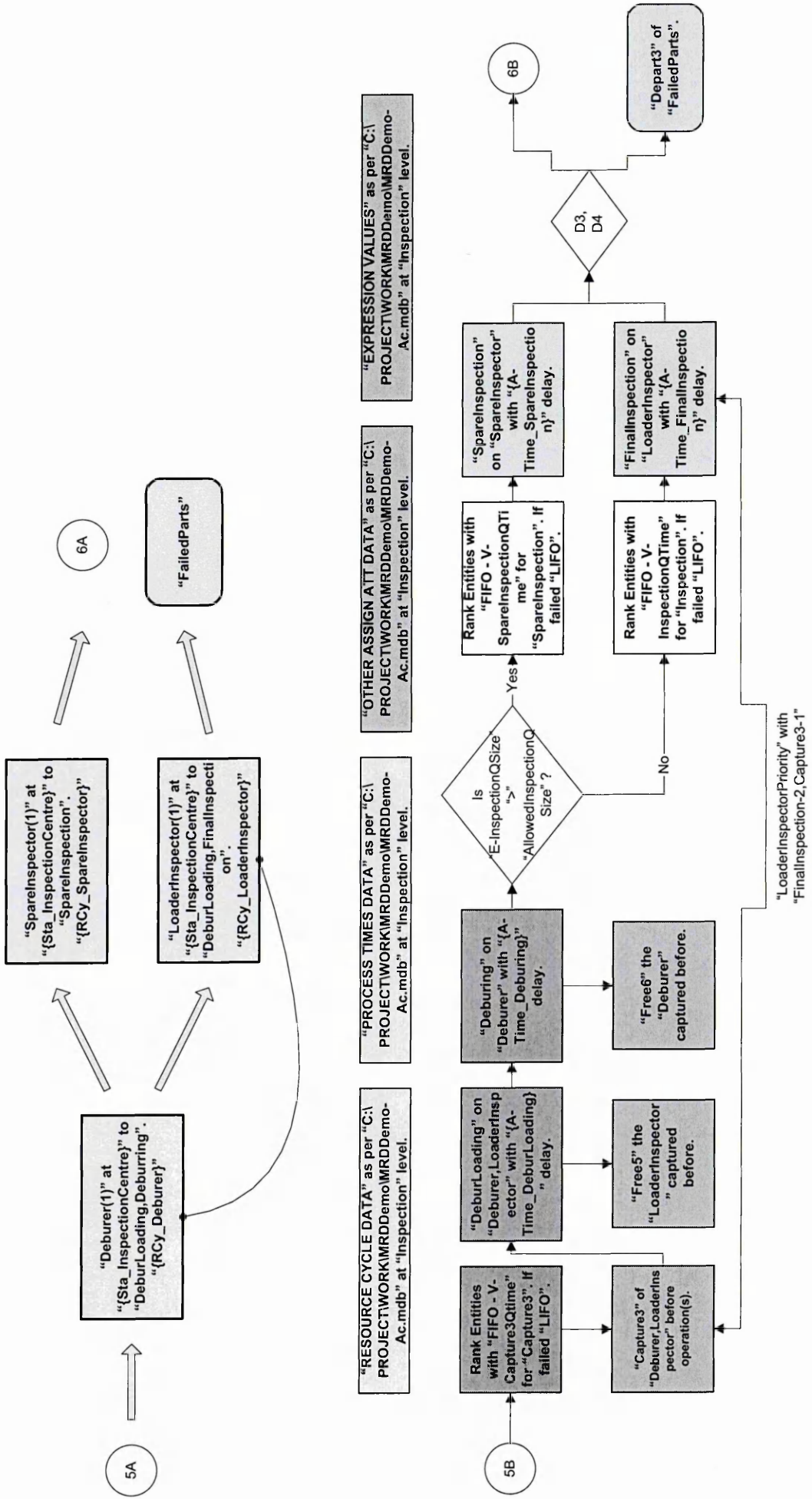
"PROCESS TIMES DATA" as per "C:\PROJECT\WORK\MRDDemo\MRDDemo-Ac.mdb" at "Departing" level.

"OTHER ASSIGN ATT DATA" as per "C:\PROJECT\WORK\MRDDemo\MRDDemo-Ac.mdb" at "Departing" level.

"EXPRESSION VALUES" as per "C:\PROJECT\WORK\MRDDemo\MRDDemo-Ac.mdb" at "Departing" level.



Sub-Model : Departing



Sub-Model : Inspection

Form 24 : Documentation Model

25. Model Data

The model data that are assigned as attributes or variables or expression and in other forms are given below.

RESOURCE CYCLE DATA									
Associated Ele Type	Associated Ele Name	RCy Name	Schedule Rule	Schedule Time Units	Schedule Values	Failure Rule	Failure Time Units	Failure Up Time	Failure Down Time
Processor	ProcessorGroup	RCy_ProcessorGroup	Ignore	Hours	1*8	Ignore	Hours	1	0
Processor	CellLoader	RCy_CellLoader	Ignore	Hours	1*8	Ignore	Hours	1	0
Processor	CNC	RCy_CNC	Ignore	Hours	1*8	Ignore	Hours	1	0
Processor	Grinder	RCy_Grinder	Ignore	Hours	1*8	Ignore	Hours	1	0
Processor	Polisher	RCy_Polisher	Ignore	Hours	1*8	Ignore	Hours	1	0
Processor	FaceMiller	RCy_FaceMiller	Ignore	Hours	1*8	Ignore	Hours	1	0
Processor	SurfaceTester	RCy_SurfaceTester	Ignore	Hours	1*8	Ignore	Hours	1	0
Processor	SurfaceHardner	RCy_SurfaceHardner	Ignore	Hours	1*8	Ignore	Hours	1	0
Processor	Miller	RCy_Miller	Ignore	Hours	1*8	Preempt	Hours	UNIF(80,85)	UNIF(4,5)
Processor	DrillB1	RCy_DrillB1	Ignore	Hours	1*8	Ignore	Hours	1	0
Processor	DrillB2	RCy_DrillB2	Ignore	Hours	1*8	Ignore	Hours	1	0
Processor	DrillBC	RCy_DrillBC	Ignore	Hours	1*8	Ignore	Hours	1	0
Processor	DrillC1	RCy_DrillC1	Ignore	Hours	1*8	Ignore	Hours	1	0
Processor	DrillC2	RCy_DrillC2	Ignore	Hours	1*8	Ignore	Hours	1	0
Processor	Borer	RCy_Borer	Ignore	Hours	1*8	Preempt	Hours	UNIF(90,100)	UNIF(6,7)
Processor	CNCGrinder	RCy_CNCGrinder	Preempt	Hours	1*8	Preempt	Hours	TRIA(87.95,100)	TRIA(6,8,10)
Processor	SpareGrinder	RCy_SpareGrinder	Preempt	Hours	1*8	Preempt	Hours	TRIA(87.95,100)	TRIA(6,8,10)
Processor	Deburer	RCy_Deburer	Preempt	Hours	1*8	Preempt	Hours	EXPO(95)	EXPO(6)
Processor	LoaderInspector	RCy_LoaderInspector	Ignore	Hours	1*8	Ignore	Hours	1	0
Processor	SpareInspector	RCy_SpareInspector	Ignore	Hours	1*8	Ignore	Hours	1	0

Resource Cycle Data

OTHER ASSIGN ATTRIBUTES					
Entity Stage Name	Associated Ele Type	Associated Ele Name	Expression	Expression Type	Data Values
PartA	Decesion	PartSeparateForGrinding	PartType	Attribute	PartTypeA
PartA	Queue	CNCGrinding.Queue	CNCGrindingPriority	Attribute	CNCGrindingPriorityA
PartB	Decesion	PartSeparateForDrilling	PartType	Attribute	PartTypeB
PartB	Queue	CNCGrinding.Queue	CNCGrindingPriority	Attribute	CNCGrindingPriorityB

OTHER ASSIGN ATTRIBUTES

Entity Stage Name	Associated Ele Type	Associated Ele Name	Expression	Expression Type	Data Values	Data Units
PartC	Decesion	PartSeparateForDrilling	PartType	Attribute	PartTypeC	
PartC	Queue	FaceMilling.Queue	FaceMillingPriority	Attribute	FaceMillingPriorityC	
PartC	Queue	Testing.Queue	TestingPriority	Attribute	TestingPriorityC	
PartC	Queue	CNCGrinding.Queue	CNCGrindingPriority	Attribute	CNCGrindingPriorityC	
ReworkPartC	Queue	FaceMilling.Queue	FaceMillingPriority	Attribute	FaceMillingPriorityRC	
ReworkPartC	Queue	Testing.Queue	TestingPriority	Attribute	TestingPriorityRC	

Other Assign Attributes

EXPRESSION VALUES

Associated Ele Type	Associated Ele Name	Expression Name	Expression Value	Description
Queue	CNCGrinding.Queue	CNCGrindingPriorityA	2	Lowest priority for PartA for CNC grinding process.
Queue	CNCGrinding.Queue	CNCGrindingPriorityB	1	Highest priority for PartB (&C) for CNC grinding process.
Queue	CNCGrinding.Queue	CNCGrindingPriorityC	1	Highest priority for PartC (&B) for CNC grinding process.
Queue	FaceMilling.Queue	FaceMillingPriorityC	2	Lowest priority for PartC for face milling process.
Queue	FaceMilling.Queue	FaceMillingPriorityRC	1	Highest priority for ReworkPartC for face milling process.
Queue	Testing.Queue	Testing PriorityC	2	Lowest priority for PartC for testing process.
Queue	Testing.Queue	Testing PriorityRC	1	Highest priority for ReworkPartC for testing process.
Decesion	PartSeparateForGr1	SpareGrindingQSize LT CNCGrindingQSize	NQ(SpareGrinding.Queue) < NQ(CNCGrinding.Queue)	Built-in variable values to be obtained from the software.
Decesion	PartSeparateForGr2	SpareGrindingQSize LE AllowedSpareGrindingQSize	NQ(SpareGrinding.Queue) <= 1	Built-in variable values to be obtained from the software.
Decesion	InspectionQSizeChecking	InspectionQSize GT AllowedInspectionQSize	NQ(Inspection.Queue) > 10	Built-in variable values to be obtained from the software.
Decesion	PartSeparateForGrinding	PartTypeA	1	Part type assigned as attribute for separation of entities.
Decesion	PartSeparateForDrilling	PartTypeB	2	Part type assigned as attribute for separation of entities.
Decesion	PartSeparateForDrilling	PartTypeC	3	Part type assigned as attribute for separation of entities.
Distribution-Out	InspectionDistribution	D3	90	% of passed parts (PartB&C) distribution at inspection process.
Distribution-Out	InspectionDistribution	D4	10	% of failed parts (PartB&C) distribution at inspection process.
Distribution-Out	TestingDistribution	D1	80	% of passed parts (PartC) distribution at testing process.
Distribution-Out	TestingDistribution	D2	10	% of failed parts (PartC) at testing; that needs rework.

Expressions

PROCESS TIME DATA

Entity Stage Name	Associated Ele Type	Associated Ele Name	Expression	Expression Type	Data Values	Data Units
PartA	Delay	ProcessDelay	Time_ProcessDelay	Attribute	EXPO(52)	Minutes
PartA	Process	CNCGrinding	Time_CNCGrinding	Attribute	UNIF(10,15)	Minutes
PartA	Process	SpareGrinding	Time_SpareGrinding	Attribute	UNIF(20,25)	Minutes
PartB	Delay	CNCLoading	Time_CNCLoading	Attribute	TRIA(2,3,4)	Minutes
PartB	Delay	CNCMachining	Time_CNCMachining	Attribute	TRIA(10,12,14)	Minutes
PartB	Delay	GrinderLoading	Time_GrinderLoading	Attribute	TRIA(2,3,4)	Minutes
PartB	Delay	Grinding	Time_Grinding	Attribute	TRIA(10,12,14)	Minutes
PartB	Process	Polishing	Time_Polishing	Attribute	TRIA(6,7,8)	Minutes
PartB	Process	Milling	Time_Milling	Attribute	EXPO(10)	Minutes
PartB	Process	DrillingB	Time_DrillingB	Attribute	EXPO(40)	Minutes
PartB	Process	Boring	Time_Boring	Attribute	EXPO(18)	Minutes
PartB	Process	CNCGrinding	Time_CNCGrinding	Attribute	UNIF(6,10)	Minutes
PartB	Process	SpareGrinding	Time_SpareGrinding	Attribute	UNIF(16,20)	Minutes
PartB	Delay	DeburLoading	Time_DeburLoading	Attribute	UNIF(3,4)	Minutes
PartB	Delay	Deburing	Time_Deburing	Attribute	UNIF(7,9)	Minutes
PartB	Process	FinalInspection	Time_FinalInspection	Attribute	EXPO(10)	Minutes
PartB	Process	SpareInspection	Time_SpareInspection	Attribute	EXPO(10)	Minutes
PartC	Process	FaceMilling	Time_FaceMilling	Attribute	TRIA(15,17,19)	Minutes
PartC	Process	Testing	Time_Testing	Attribute	TRIA(15,17,19)	Minutes
PartC	Process	Milling	Time_Milling	Attribute	EXPO(15)	Minutes
PartC	Process	DrillingC	Time_DrillingC	Attribute	EXPO(35)	Minutes
PartC	Process	Boring	Time_Boring	Attribute	EXPO(20)	Minutes
PartC	Process	CNCGrinding	Time_CNCGrinding	Attribute	UNIF(12,16)	Minutes
PartC	Process	SpareGrinding	Time_SpareGrinding	Attribute	UNIF(22,26)	Minutes
PartC	Delay	DeburLoading	Time_DeburLoading	Attribute	UNIF(5,7)	Minutes
PartC	Delay	Deburing	Time_Deburing	Attribute	UNIF(7,9)	Minutes
PartC	Process	FinalInspection	Time_FinalInspection	Attribute	EXPO(10)	Minutes
ReworkPartC	Process	Hardning	Time_Hardning	Attribute	TRIA(2,3,4)	Minutes
ReworkPartC	Process	FaceMilling	Time_FaceMilling	Attribute	TRIA(2,3,4)	Minutes
ReworkPartC	Process	Testing	Time_Testing	Attribute	TRIA(2,3,4)	Minutes

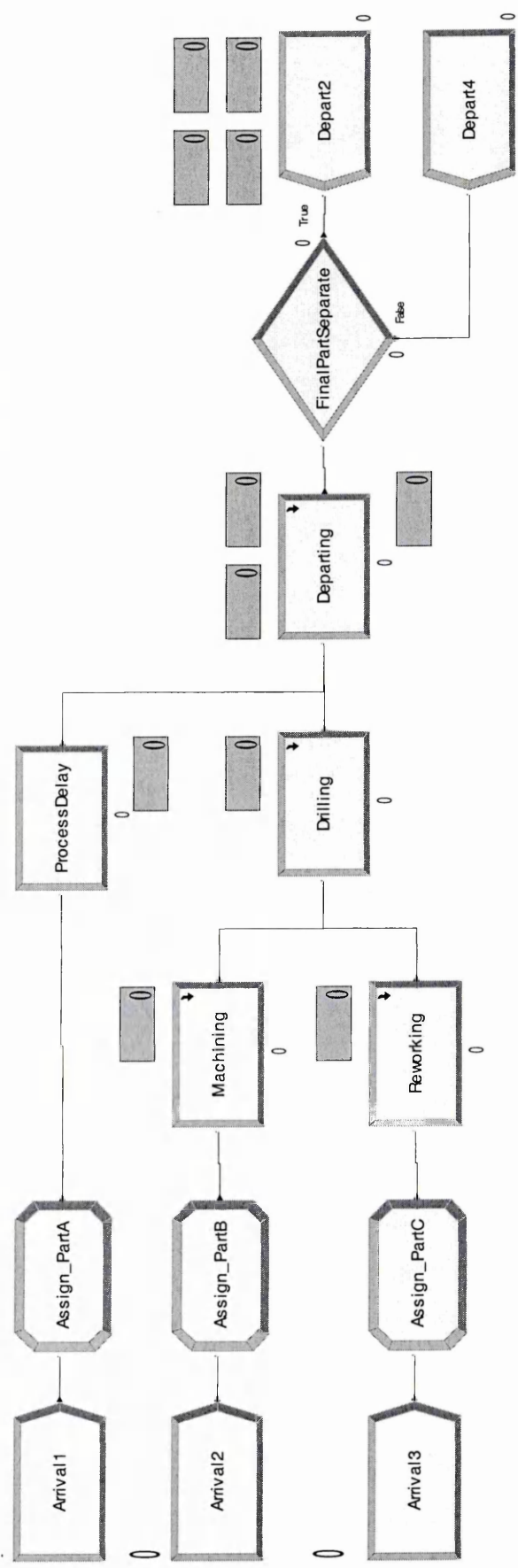
Process Time

Form 25 : Model Data

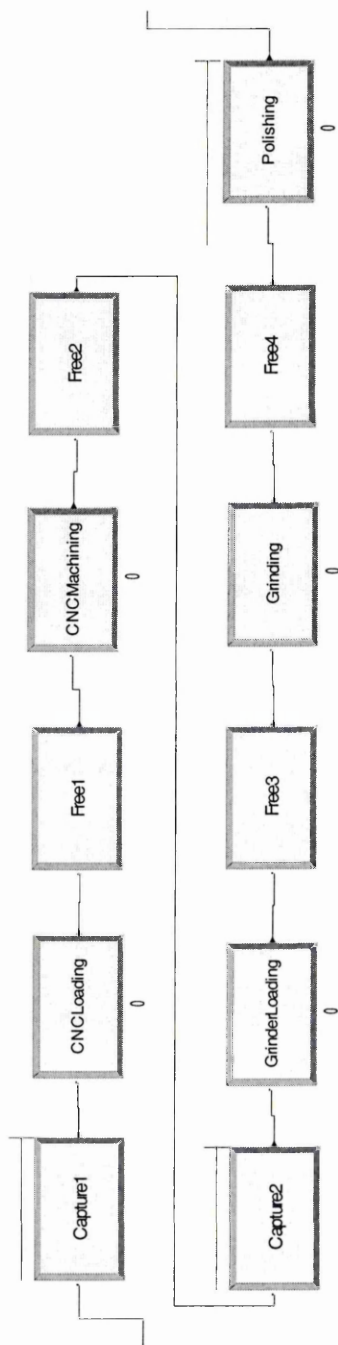
26. Model Details

The form below shows what software was used for the model, where the model & associated details available, and the other potential details either on both in graphical or text forms.

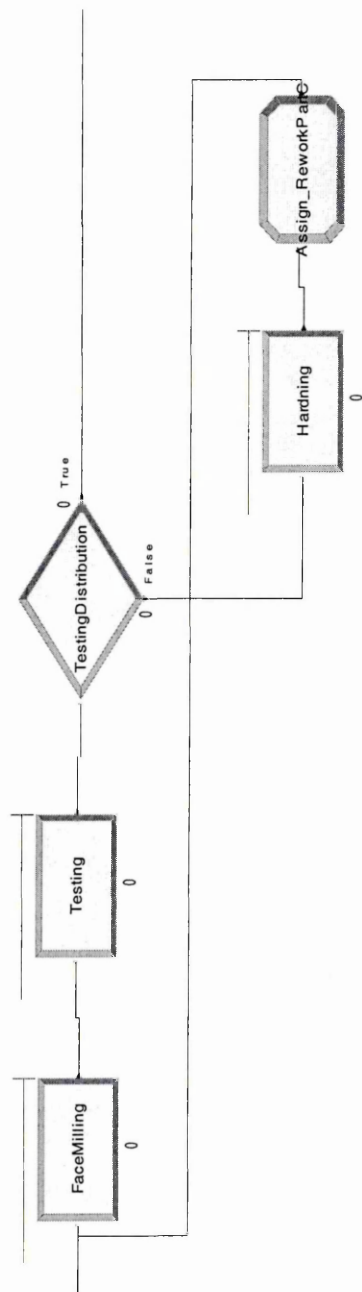
SIMULATION SOFTWARE	:	Arena	MODEL FILE NAME	:	MRDDemo-Ar
OTHER SOFTWARE (If, available)	:	None.	SUPPLEMENTARY DATA FILENAME (If, available)	:	None.
MODEL CODES	:	See below for graphical model in the simulation software.			
ANY SUPPLEMENTARY DETAILS:	:	None.			
References/ Appendix	:	Affixed for demonstration purpose.			



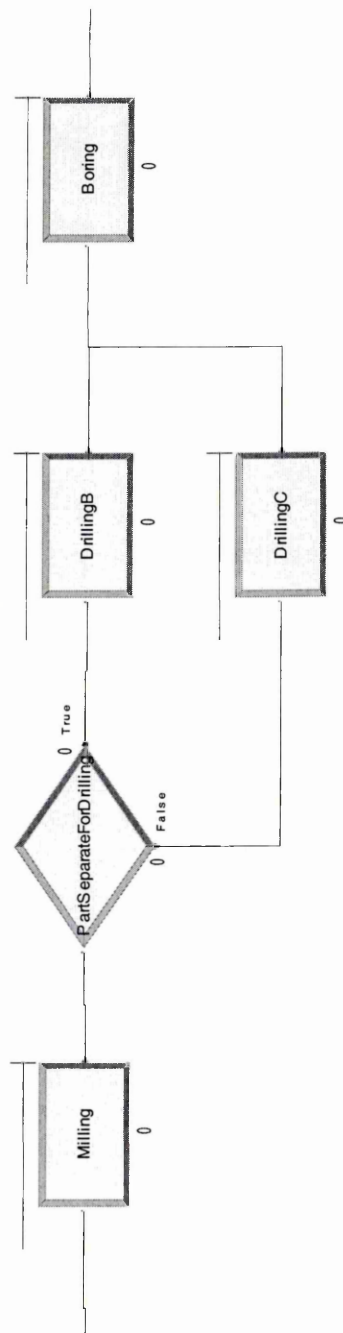
Overall Model



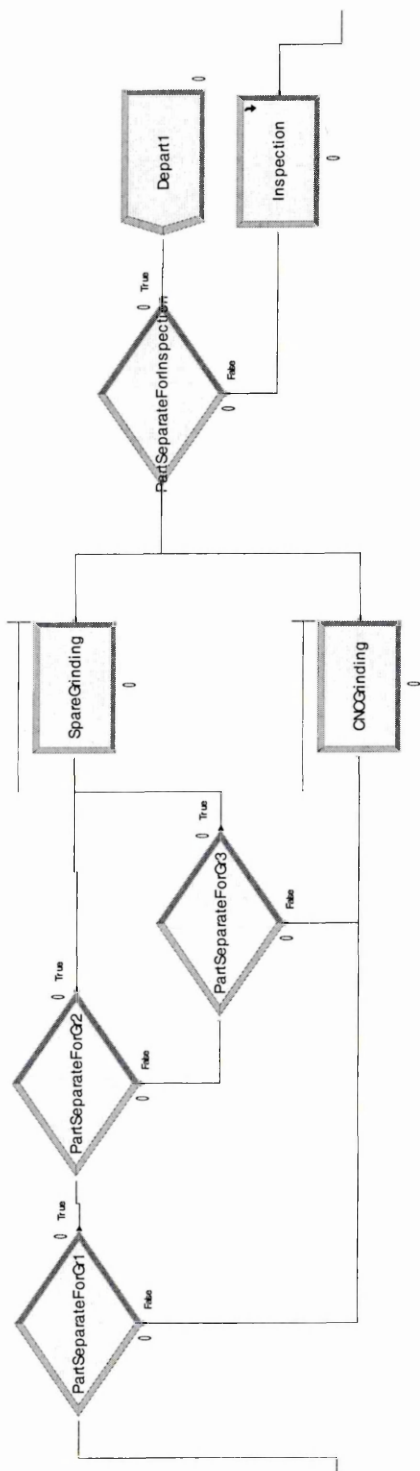
Sub Model : Machining



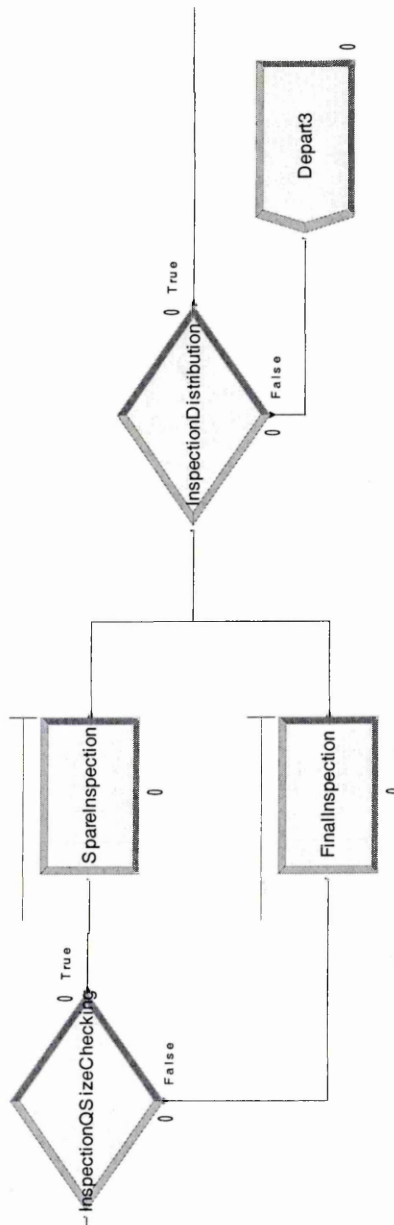
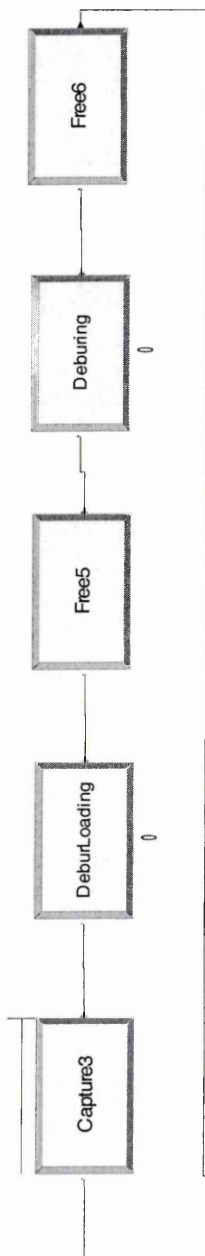
Sub-Model : Reworking



Sub Model : Drilling



Sub Model : Departing



Sub Model : Inspection

Form 26 : Model Details

27. Verification Details

The table below presents what elements and how the individual elements are verified.

VERIFICATION LIST (ALL ELE)			
Element Type	Element Name	Verification Techniques Employed	Verification Process and Output Details
Arrival	Arrival1	1,4	Need alteration to time between arrivals.
Delay	ProcessDelay	3	Compatible with realistic output. See Annex 3.
Decesion	PartSeparateForGr1	2,4	Behave as expected.
Decesion	PartSeparateForGr2	2,4	Neer adjustment to allowed queue size.
Decesion	PartSeparateForGr3	2,4	Behave as expected.
Ranking	CNCGrinding.Queue	4	Display real queue size.
Process	CNCGrinding	1	Agreed.
Processor	CNCGrinder	3,4	Behaves realistically, including the failure.
Ranking	SpareGrinding.Queue	3	Compatible with realistic output. See Annex 3.
Process	SpareGrinding	1	Agreed.
Processor	SpareGrinder	3	Output seems to correct compare with the output of CNC grinder. See Annex 3.
Decesion	PartSeparateForInspection	4	Behave as expected.
Ranking	Capture3.Queue	2	Compatible with realistic output. See Annex 3.
Capture	Capture3	4	Behaves realistically.
Priority	LoaderInspectorPriority	1,4	Created graphical environment to see the priority effect.
Priority	LoaderInspectorPriority	1,4	Created graphical environment to see the priority effect.
Priority	LoaderInspectorPriority	1,4	Created graphical environment to see the priority effect.
Set	DrillBSet	1,2	Code check for order of members.
Selection	SEL_DrillBSet	1,4	Created graphical environment to see the priority effect.
Submodel	Rework	3,4	Created a dummy entity to check the behaviour and out put; and output is realistic.
Submodel	Machining	3	Created a dummy entity to check the behaviour and out put; and output is realistic.
Techniques		References/Appendix	
KEYS *	1. Walk-through with an experts	Annex 3 (verification data) (but, not attached in the example.)	
	2. Visual check with system experts.		
	3. Comparison with expected results.		
	4. Animation		

Form 27 : Verification Details (Part of the List)

28. Details in Operational Validity

The following form displays how the model is validated and what assurance is given about the working model.

VALIDATION TECHNIQUES EMPLOYED	PROCESS & DATA DESCRIPTION IN VALIDATION	RESULTS OBTAINED AND CONCLUSION
Statistical and animation.	Throughput and time of machining sub-model.	Average throughput is 55.2 units per week in 4 replications. It is about ± 5 real values that may be due to transportation. (Results in Annex 4). Valid sub-model.
Statistical and animation.	Throughput and time of reworking sub-model.	Average throughput is 57.1 units per week in 4 replications. It is about ± 6 real values that may be due to transportation. (Results in Annex 4). Valid sub-model.
Statistical and animation.	Throughput and time of drilling sub-model.	Average throughput is 42.8 units per week in 4 replications. It is about ± 5 real values that may be due to transportation. (Results in Annex 4). Valid sub-model.
Statistical and animation.	Throughput and time of inspection sub-model.	Average throughput is 61.9 units per week in 4 replications. It is about ± 4 real values that may be due to transportation. (Results in Annex 4). Valid sub-model.
Statistical and animation.	Throughput and time of overall model.	Average throughput is 59.4 units per week in 4 replications. It is about ± 5 real values that may be due to transportation. (Results in Annex 4). Valid model.
References/Appendix :	Annex 4 : Validation details (but not included in the example).	

Form 28 : Details in Operational Validity

29. Details in Parameter Validity (Sensitivity Analysis)

The table below presents what data items involved with sensitivity analysis; what assurance on their accuracy and what results are affected in failing to achieve expected accuracy.

DATA ITEM ANALYSED	BRIEF DESCRIPTION ON ANALYSIS PROCESS	RESULTS OBTAINED, CONCLUSION MADE, ANY RESULTS BE AFFECTED
No Sensitivity analysis is performed.	Not Applicable.	Not Applicable.
References/Appendix :	Not Applicable.	

Form 29 : Details in Parameter Validity (Sensitivity Analysis)

30. General Overview on Experimentation

The table below gives you an idea about the model characteristics, model behaviour, and experimentation approaches in a model run.

MODEL TYPE:	True Deterministic :	No	Stochastic :	Yes
MODEL END SITUATION:	Terminating :	No	Non-Terminating:	Yes
SIMULATION STATUS:	Steady-State Cycle :	Yes	Steady-State :	No
EXPERIMENT APPROCH(ES):	Interactive :	No	Batch :	Yes
	Comparative :	Yes	Predictive :	No
ADDITIONAL NOTES :	None.			

Form 30 : General Overview on Experimentation

31. Warm-Up Period (WUP) and Starting Conditions

The form below depicts what stating condition that the model has, how the staring condition is to be established, how theoretical WUP designed and what data involved in starting conditions and in designing WUP.

STARTING CONDITIONS:	Zero Starting Condition :	No	Variable Starting Condition :	No
Data and Description	:	Not Applicable.		
WARM-UP PERIOD:	Visual Inspection :	Yes	Time Series Analysis :	No
Theoretical Value & Units:	:	Around 3.3 hours.		
Data and Description	:	Visual inspection in 3 replications displays that the system becomes realistic steady state in just above three hours time.		
References/Appendix	:	None.		

Form 31 : Warm-Up Period and Starting Condition

32. Run Length

The table below presents approximate value run length and based events; how run length was designed and its designed value with data description.

APPROX. VALUE & UNITS IN EVENT OBSERVATION :	Not applicable	LIST OF INFREQUENT EVENTS :	Not applicable
THEORATI. VALUE & UNITS :	Not applicable	% OF CONFIDENCE :	Not applicable
DATA AND DESCRIPTION :	Steady state cycle – due to regular pattern of arrival in each week; thus the run length is one week.		
References/Appendix :	Not applicable		

Form 32 : Run Length

33. Number of Replications

The form below illustrates the theoretical value for number of replications with their confidence level and data involved with design process.

THEORATI. VALUE & UNITS :	Not designed.	% OF CONFIDENCE :	Not applicable
DATA AND DESCRIPTION :	Not conducted due to time constraint. General rule of thumb:3-5 replications were taken into account.		
References/Appendix :	Not applicable		

Form 33 : Number of Replications

34. Summary of Experiment Run

The table below summarizes the details of experiment run, containing experiments conducted, their factor levels, experiment approach and the practiced values on designed parameters such as WUP, run length and number of replications.

EXPERIMENT DETAILS	NO. OF FACTOR LEVELS PRACTICED	PRACTICED WARM-UP PERIOD & UNITS	PRACTICED RUN LENGTH & UNITS	NO. OF REPLICATIONS PRACTICED	EXPERIMENT APPROACH	USEFUL NOTES
1- Throughput	Increase in milling machine capacity	4 hours	Week	4	Comparative	WUP is considered as 4 hours & run length (steady-state cycle) is treated as 5 normal days
2- Throughput	Increase in boring machine capacity	4 hours	Week	4	Comparative	

3- Throughput	Increase in CNC grinder capacity	4 hours	Week	4	Comparative	WUP is considered as 3 hours & run length (steady-state cycle) is treated as 5 normal days
4- Throughput	Addition of Spare grinder for grinding	4 hours	Week	4	Comparative	
5- Throughput	Addition Spare grinder & Increase in miller capacity	4 hours	Week	4	Comparative	
6- Throughput	Addition Spare grinder & Increase in borer capacity	4 hours	Week	4	Comparative	
7- Throughput	Introduction of shift to drilling centre	4 hours	Week	4	Comparative	
8- Throughput	Introduction of shift to departing centre	4 hours	Week	4	Comparative	
9- Throughput	Introduction of shift to drilling and departing centre	4 hours	Week	4	Comparative	

Form 34 : Summary of Experiment Run

35. Summary of Results

Summary of results of experiments or output of analysis process are briefed in the following table.

EXPERIMENT										RESULTS	
No.	FACTOR 1		FACTOR 2		FACTOR 3		FACTOR 4		REP. NO. /NO. RANGE	Part A + Part B + Part C	Appendix/ Reference
	NAME	VALUE	NAME	VALUE	NAME	VALUE	NAME	VALUE			
1	Miller Cap	2	Borer Cap	1	CNCGr cap	1	Spare Gri	0	1-4	182	Annex 5 (not included)
2	Miller Cap	1	Borer Cap	2	CNCGr cap	1	Spare Gri	0	1-4	194	Annex 5 (not included)
3	Miller Cap	1	Borer Cap	1	CNCGr cap	2	Spare Gri	0	1-4	214	Annex 5 (not included)
4	Miller Cap	1	Borer Cap	1	CNCGr cap	1	Spare Gri	1	1-4	245	Annex 5 (not included)
5	Miller Cap	2	Borer Cap	1	CNCGr cap	1	Spare Gri	1	1-4	251	Annex 5 (not included)
6	Miller Cap	1	Borer Cap	2	CNCGr cap	1	Spare Gri	1	1-4	273	Annex 5 (not included)
7	Drill Shift	2	Depar Shift	1	---	---	---	---	1-4	193	Annex 5 (not included)
8	Drill Shift	1	Depar Shift	2	---	---	---	---	1-4	236	Annex 5 (not included)
9	Drill Shift	2	Depar Shift	2	---	---	---	---	1-4	243	Annex 5 (not included)

Form 35 : Summary of Results

36. Conclusions and Recommendations

The conclusions drawn and recommendations made with details associated in reckoning process are given in the following table.

CONCLUSIONS :	
1.	<i>Introduction of spare grinder at departing centre alone significantly increases the throughput.</i>
2	<i>Introduction of boring machine alone at drilling centre decreases the throughput (total of part A, B, C), due prioritizing the product for CNC grinding.</i>
3.	<i>Introduction of milling machine alone at drilling centre has no much effect to throughput.</i>
4.	<i>Introduction of milling machine with spare grinder increases the throughput, but the increase is negligible compare with conclusion 1.</i>
5.	<i>Introduction of boring machine with spare grinder increases the throughput marginally, making it difficult to ignore.</i>
6.	<i>Increase in shift does not have much influence on throughput time, instead machine utilization become low.</i>
7.	<i>Introduction of spare inspector would be waste due to no usage of the resources..</i>
RECOMMENDATIONS:	
1.	<i>Introduce the spare grinder for grinding operation.</i>
2.	<i>Increase the capacity of boring machine.</i>
OTHER DETAILS :	
<i>None.</i>	

Form 36 : Conclusions and Recommendations

37. Review of the Results

The form below delivers what and how the results were presented and evaluated, who participated for the review and what feedback received and level of recommendations are agreed.

WRITTEN REPORTS :	Yes	PRESENTATION AND/OR DEMONSTRATION :	Yes
PARTICIPANTS FOR REVIEW :	Simulation project team, managers in production planning and supply divisions, engineers in manufacturing and assembly divisions; and implementation team.		
LIST OF DETAILS REPORTED/ PRESENTED :	Problem objectives; experiment details; report details; documentation model; details in operational validity; summary of experiment run; summary of results; conclusions and recommendations.		
RECEIVED FEEDBACK :	Yes	AGREED LEVEL OF RECOMMENDATIONS :	Yes
ANY ALTERATIONS TO RECOMMENDATIONS :	None.		

Form 37 : Review of the Results

38. Review of the Project

The project teams' and client's perception about the project are presented in the following form.

STATUS OF THE PROBLEM OBJECTIVES :	
Objectives, met :	Increase of throughput with introduction of spare grinder, boring machine and additional shift.
Objectives, not met :	Opportunity to study performance or resources in full. This was due to time constraint.
PERSPECTIVES OF THE PROJECT TEAM :	
Overall performance :	Satisfied
What has been done well? :	Documentation and executable model.
What should have been done better? :	Experimentation.
What needs to be improved later? :	Design of experimental plan.
PERSPECTIVES OF THE CLIENT :	
Overall Satisfaction :	Satisfied
What has been achieved well? :	The model within the specified time.

What should have been achieved better?: *Interaction with more details for studying the system.*

ADDITIONAL NOTES : *Project is not yet implemented to comment on the successfulness completely.*

Form 38 : Review of the Project

39. Implementation Plan

The following records deliver who will implement the project, what and how the recommendations are practiced and in what stage the progress need to be monitored.

RECOMMENDATIONS TO BE PRACTICED :

1. *Introduction of Spare Grinder for CNC grinding operation enabling to increase the throughput by 20% and later expand departing centre for to provide service for Parts B and C only with CNC grinder.*
2. *Increase of capacity of the boring machine that increases the throughput time by 6%.*

MEMBERS OF IMPLEMENTATION TEAM :

Model executor (from simulation team) and other nominated members by the Client.

	Stages	Completion/Due Date
IMPLEMENTATION :	<i>Introduction of Spare grinder</i>	<i>10th May 2004</i>
	<i>Introduction of boring machine</i>	<i>30th May 2004</i>
MONITORING :	<i>Introduction of Spare grinder</i>	<i>24th May 2004</i>
	<i>Introduction of boring machine</i>	<i>21st June 004</i>

Form 39 : Implementation Plan

40. Documents Prepared

The form below presents the current status of the project, where the complete document available for later use and what documents are prepared.

DOCUMENT FILE NAME (S) :	<i>C:/PROJECT/WORK/MRDDemoAll.Doc</i>	CURRENT STATUS OF THE PROJECT :	<i>Implementation stage.</i>				
DOCUMENTS PREPARED :	<table> <tr> <th>Audience</th> <th>Purpose</th> </tr> <tr> <td><i>Full document (all phase) only.</i></td> <td><i>To track the project progress traceable.</i></td> </tr> </table>	Audience	Purpose	<i>Full document (all phase) only.</i>	<i>To track the project progress traceable.</i>		
Audience	Purpose						
<i>Full document (all phase) only.</i>	<i>To track the project progress traceable.</i>						

Form 40 : Documents Prepared

ANNEXURE "D" – SUMMARY OF THE QUESTIONNAIRE SURVEY RESULTS

DESCRIPTION	Scheme	Respondent												Total	Percent (%)	Average	Concluding Remarks
		1	2	3	4	5	6	7	8	9	10	11	12				
ROLE OF EXPRISE IN SIMULATION (Q. 1.1)																	
Academic/Researcher	1					1	1	1	1	1	1	1	1	7	58%	5/12	Survey represents both academic & industrial community
Academic & Industrial	0			0	0					0				0	25%	=	
Industrial	-1	-1												-2	17%	0.42	
NO. OF SIMULATION MODELS DEVELOPED (Q. 1.2)																	
None	0									0				0	00%	18/12	Participants have experience with modelling more than 5 models.
1 to 5	1	1			1	1					1		4	33%	=		
> 5	2	2	2	2			2		2	2	2		14	58%	1.50		
SIMULATOR(S) THAT MORE EXPERIENCE WITH (Q. 1.3)																	
Arena		x	x	x	x	x			x		x	x	10x	83%		Most of the participants have experience with Arena.	
Quest			x										1x	08%			
ProModel				x	x								2x	17%			
Witness							x						2x	17%			
Simul8							x		x				2x	17%			
Other software				VenSim/VenSim			ATOMS				MPlant						
ARE THE DETAILS GENERALLY RECORDED? (Q. 2.1)																	
Always Yes	4			4									4	8%	30/12	General practice is that the details are usually recorded.	
Usually Yes	3	3	3		3				3		3		15	42%	=		
Sometimes Yes	2			2			2	2		2		2	10	42%	2.50		
Seldom Yes	1						1						1	08%			
Always No	0												0	00%			
No Experience	0												0	00%			
WHAT LEVEL OF DETAILS IS GENERALLY RECORDED? [Q. 2.2(a)]																	
Comprehensive details for all audiences	1	1											1	08%	1/12	Recording moderate details is the usual practice.	
Moderate details for modelling, run, etc.	0		0	0	0	0	0	0	0	0	0	0	0	92%	=		
Minimal details for client	-1												0	00%	0.08		
HOW THE DETAILS ARE GENERALLY RECORDED? [Q. 2.2(b)]																	
Progressively (as project progresses)	1									1	1		2	17%	1/12	Generally the details are recorded progressively & retrospectively	
Progressively & Retrospectively	0	0	0	0	0		0	0	0		0	0	0	75%	=		
Retrospectively (after project)	-1				-1								-1	08%	0.08		
Always independently from simulator	1												0	00%	-1/12	Not a structured question. Seems, details are both s/w dependent & independent.	
Usually independently from simulator	0						0		0				0	17%	=		
Mostly depend on simulator	-1											-1	-1	08%	-0.08		
(No Answer to S/W independent details)		x	x	x	x	x	x		x		x	x		75%			

FACTORS THAT ARE OBSTRUCTED FOR DOCUMENTATION [Q. 2.2(c)]																			
Difficulty in identification of details				x	x	x											x		
No structured methodology				x	x	x											x		
Isolated process from modelling					x	x												x	
No time for proper documentation				x	x														
Forget details in retrospective doc'n				x															
CHARACTERISTICS TO BE INCORPORATED IN DOCUMENTATION PROCESS (Q. 2.4)																			
Documentation should be progressive with model the development process																			
• Strongly agree	2			2	2	2				2									
• Agree	1									1									
• Neither Agree/Disagree	0	0																	
• Disagree	-1																		
• Strongly disagree	-2																		
Documentation should be independent from the simulation software																			
• Strongly agree	2									2									
• Agree	1					1	1												
• Neither Agree/Disagree	0	0																	
• Disagree	-1																		
• Strongly disagree	-2																		
Documentation system should have common layout with structured and linked details																			
• Strongly agree	2									2									
• Agree	1	1	1	1	1	1													
• Neither Agree/Disagree	0																		
• Disagree	-1																		
• Strongly disagree	-2																		
Documentation process should be integrated with the model development process																			
• Strongly agree	2					2	2	2	2										
• Agree	1	1	1																
• Neither Agree/Disagree	0																		
• Disagree	-1																		
• Strongly disagree	-2																		
IS THE PROPOSED DEFINITION FOR DOCUMENTATION NECESSARY? (Q. 3.1)																			
Definitely Yes	2									2									
Yes	1				1	1	1												
NOT Sure	0																		
No	-1																		
Necessity of explicit definition for doc'n (MRD) is accepted.																			

IS THE TASK-ORIENTATION APPROPRIATE TO FULFIL THE DIFFERENT PURPOSES IN DOCUMENTATION? (Q. 3.3)													10/12	Agree that Task-orientation is appropriate to fulfil purposes of doc'n (MRD).	
Strongly agree	2												0	00%	0.83
Agree	1	1	1	1	1	1	1	1	1	1	1	1	10	83%	
Neither Agree/Disagree	0												0	17%	
Disagree	-1												0	00%	
Strongly disagree	-2												0	00%	
DOES THE USE OF TYPICAL (REVISED) PROJECT PROCEDURE REPRESENT A GENERIC FRAMEWORK FOR TASK-ORIENTATION? (Q. 3.4)															
Definitely Yes	2												0	00%	0.75
Yes	1	1	1	1	1	1	1	1	1	1	1	1	9	75%	
NOT Sure	0												0	25%	
No	-1												0	00%	
													0	00%	
BENEFITS EXPECTED FROM TASK DOCUMENTATION AND TASK DOCUMENTS (Q. 3.5)															
Task documents serve as a check list for documentation and model development process															
Strongly agree	2												8	33%	1.33
Agree	1	1	1	1	1	1	1	1	1	1	1	1	8	67%	
Neither Agree/Disagree	0												0	00%	
Disagree	-1												0	00%	
Strongly disagree	-2												0	00%	
Pre-structured forms in task documents provides uniform structure for recording task details															
Strongly agree	2												8	34%	1.25
Agree	1	1	1	1	1	1	1	1	1	1	1	1	7	58%	
Neither Agree/Disagree	0	0											0	08%	
Disagree	-1												0	00%	
Strongly disagree	-2												0	00%	
Pre-structured forms encourage/help modellers for documentation															
Strongly agree	2												4	17%	1.17
Agree	1	1	1	1	1	1	1	1	1	1	1	1	10	83%	
Neither Agree/Disagree	0												0	00%	
Disagree	-1												0	00%	
Strongly disagree	-2												0	00%	
Pre-structured forms guide the modellers for performing tasks in simulation projects															
Strongly agree	2												4	16.5%	1.00
Agree	1												8	67%	
Neither Agree/Disagree	0	0											0	16.5%	
Disagree	-1												0	00%	
Strongly disagree	-2												0	00%	

Pre-structured forms lay a sound foundation for integrated documentation system												
• Strongly agree	2	1	1	1	1	2	2	1	1	4	17%	11/12 = 0.92
• Agree	1	1	1	1	1			1	1	7	58%	
• Neither Agree/Disagree	0					0	0		0	0	25%	
• Disagree	-1									0	00%	
• Strongly disagree	-2									0	00%	
DOES REUSABLE MODEL ELEMENTS REPRESENT SOFTWARE INDEPENDENT TASK DETAILS? (Q. 3.6)												
Definitely Yes	2									0	00%	4/12 = 0.33
Yes	1	1			1	1	1		1	5	42%	
NOT Sure	0	0	0	0	0			0	0	0	50%	
No	-1									-1	08%	
DOES REUSABLE MODEL ELEMENTS DESCRIBE THE MODEL EXPLICITLY? (Q. 3.7)												
Definitely Yes	2		2	2						2	25%	13/12 = 1.08
Yes	1		1		1	1	1	1	1	8	67%	
NOT Sure	0									0	00%	
No	-1	-1								-1	08%	
DOES AMMULGAMATED REUSABLE MODEL ELEMENTS AND FLOWCHART SYMBOLS DESCRIBE THE MODEL INDEPENDENTLY? (Q. 3.8)												
Definitely Yes	2									0	00%	3/12 = 0.25
Yes	1	1		1				1	1	5	42%	
NOT Sure	0		0	0	0	0	0		0	0	42%	
No	-1	-1								-2	16%	
DO COMMON TEMPLATES ENCOURAGE/HELP MODELLERS FOR DOCUMENTATION? (Q. 4.1)												
Definitely Yes	2					2	2			4	17%	14/12 = 1.17
Yes	1	1	1	1	1		1	1	1	10	83%	
NOT Sure	0									0	00%	
No	-1									0	00%	
DO THE LINKED DETAILS ENCOURAGE/HELP MODELLERS FOR DOCUMENTATION? (Q. 4.2)												
Definitely Yes	2									0	00%	9/12 = 0.75
Yes	1		1	1	1	1	1	1	1	9	75%	
NOT Sure	0	0	0				0			0	25%	
No	-1									0	00%	
CAPABILITIES TO BE INCORPORATED FOR HANDLING TAK DOCUMENTS IN MRD SYSTEM (Q. 4.3)												
Display different stages of the project		x	x	x	x	x	x	x	x	8x	67%	1 (display stages)
Easy accessible to TD at any time			x	x	x	x	x	x	x	8x	67%	1 (easy access TD)
Immediate display for task details			x	x	x	x		x	x	7x	58%	2 (imm display TD)
Alteration with least interaction							x	x	x	4x	33%	3 (least interaction)
Display recorded/unrecorded TDs				x	x	x	x	x	x	7x	58%	2 (display recorded)
Highlights TDs that need alterations			x	x	x	x	x	x	x	8x	67%	1 (highlight TD)
Other capabilities												Project management facility

OTHER FACILITIES TO BE INCORPORATED IN MRD SYSTEM (Q. 4.5)														
Sharing Details among S/W for docu'n														
Produce document in any stage														
Other facilities	Flexible	Web	Web	Web	PoPoint	Integr								
2 (communicate with s/w in doc'n)														
1 (produce doc's in any stage)														
Access to Web; Power Point pres'n														
DOES THE INTEGRATED MRD SYSTEM ENCOURAGE/HELP MODELLERS FOR DOCUMENTATION? (Q. 5.1(a))														
Definitely Yes	2				2	2	2	2	2	2	2	2	2	2
Yes	1	1	1	1	1	1	1	1	1	1	1	1	1	1
NOT Sure	0													
No	-1													
DOES COMMON DATABASE LAY A FOUNDATION FOR MODEL REUSABILITY? (Q. 5.1(b))														
Definitely Yes	2		2		2	2	2	2	2	2	2	2	2	2
Yes	1		1	1	1	1	1	1	1	1	1	1	1	1
NOT Sure	0				0									
No	-1													
DOES THE COMMON DATABASE PROPOSED MAKES PROVISION FOR MODEL TRANSLATION? (Q. 5.1(c))														
Definitely Yes	2													
Yes	1	1			1	1	1	1	1	1	1	1	1	1
NOT Sure	0		0	0	0	0	0	0	0	0	0	0	0	0
No	-1													
IS THE STANDRD MRD SYSTEM IS VIABLE AND PRACTICABLE? (Q. 5.2)														
Definitely Yes	2													
Yes	1	1	1	1	1	1	1	1	1	1	1	1	1	1
NOT Sure	0													
No	-1													
OVERALL ASSESSMENT ON PROPOSED MRD PROCESS AND SYSTEM (Q. 6.1)														
MRD Process is time consuming than the normal practice														
Strongly agree	2	2												
Agree	1													
Neither Agree/Disagree	0		0											
Disagree	-1			-1										
Strongly disagree	-2													
MRD Process is lengthier than the normal practice														
Strongly agree	2	2												
Agree	1													
Neither Agree/Disagree	0		0											
Disagree	-1			-1										
Strongly disagree	-2													
OVERALL ASSESSMENT ON PROPOSED MRD PROCESS AND SYSTEM (Q. 6.1)														
MRD Process is time consuming than the normal practice														
Strongly agree	2	2												
Agree	1													
Neither Agree/Disagree	0		0											
Disagree	-1			-1										
Strongly disagree	-2													
MRD Process is lengthier than the normal practice														
Strongly agree	2	2												
Agree	1													
Neither Agree/Disagree	0		0											
Disagree	-1			-1										
Strongly disagree	-2													

MRD process is useful for the success of simulation projects and enhances model reuse														10/12		Agree that MRD process is useful for the success of project & enhance model reuse.
• Strongly agree	2			2	2									6	25%	
• Agree	1	1	1		1							1		5	42%	
• Strongly disagree	0											0		0	16.5%	
• Disagree	-1											-1		-1	16.5%	
• Neither Agree/Disagree	-2													0	00%	
MRD system encourages/helps modellers for documentation of simulation projects														14/12		Agree that MRD system helps modellers for documentation of sim projects.
• Strongly agree	2		2											4	17%	
• Agree	1	1		1	1							1	1	10	83%	
• Neither Agree/Disagree	0													0	00%	
• Disagree	-1													0	00%	
• Strongly disagree	-2													0	00%	
MRD system acts as management, control and monitoring system of simulation projects														14/12		Agree that MRD system acts as a mgt, control and monitoring system of sim projects.
• Strongly agree	2		2									2		4	17%	
• Agree	1	1		1	1	1	1	1	1	1	1	1	1	9	83%	
• Neither Agree/Disagree	0													0	00%	
• Disagree	-1													0	00%	
• Strongly disagree	-2													0	00%	
MRD system makes the documentation to consider as an integral activity in a simulation project														10/12		Ambiguous question. MRD process may consider as an integral activity.
• Strongly agree	2											2		2	08%	
• Agree	1			1	1	1						1	1	8	67%	
• Neither Agree/Disagree	0	0	0						0					0	25%	
• Disagree	-1													0	00%	
• Strongly disagree	-2													0	00%	
MRD system acts as a training system to novice user														12/12		Agree that MRD system acts as a training system for a novice user.
• Strongly agree	2											2		4	16.5%	
• Agree	1	1			1	1						1	1	8	67%	
• Neither Agree/Disagree	0		0						0					0	16.5%	
• Disagree	-1													0	00%	
• Strongly disagree	-2													0	00%	
Proposed documentation model and common database act as a media for model reusability														9/12		Not a structured question. DM and common database may enhance reusability.
• Strongly agree	2													0	00%	
• Agree	1		1									1	1	9	75%	
• Neither Agree/Disagree	0	0			0	0								0	25%	
• Disagree	-1													0	00%	
• Strongly disagree	-2													0	00%	

Experience with.; model exchange (Q. 6.2)									
Arena	Arena Quest	Pro Model	Pro Model	Arena	Arena	Witness Simulac	Arena	Witness	Arena
Highly likely						x			
Likely	x			x				x	x
Unlikely									
Highly – unlikely									
NO Comment		x	x	x			x		
OTHER COMMENTS (Q. 7)									
A	1.A		3.A	4.A	6.A	7.A		9.A	11.A
B	1.B		3.B	4.B	6.B	7.B		9.B	11.B
C	1.C		3.C	4.C		7.C		9.C	
D	1.D		3.D	4.D		7.D			
E	1.E					7.E			
1.A	Pro'd type of MRD is very good & necessary.			3.A	Useful for shortening the project duration			4.A	Useful for shortening the project duration
1.B	Make documentation a lot easier.			3.B	Improve communication within the team			4.B	Improve communication within the team
1.C	Increase value of simulation projects			3.C	Hard to join with different packages			4.C	Hard to join with different packages
1.D	Doubtful – length; too much detail.			3.D	Requires universal knowledge repository			4.D	Requires universal knowledge repository
1.E	Probably refused by the client.			3.E				4.E	
6.A	Interesting to have a traceability (version control)			7.A	Planning phase tends to single plan			9.A	Doc'n – important, generally done poorly
6.B	Unable track changes.			7.B	Need to able to trace the progress			9.B	Generic approach not possible due to different problem domain, s/w and mode of practice
				7.C	Usual practice: CM, before scope & level				
				7.D	Avoid duplication of data (CM & run model)				
11.A	Lot of extra work.			7.E	Results format – insufficient, need link 35&10			9.C	Doc'n – personal thing; thus difficult to standardize, except general guide lines
11.B	Document only important details.								